A.I. in the Office and the Factory: Evidence from Administrative Software Registry Data*

Gustavo de Souza†

January, 2026

Abstract

What is the effect of artificial intelligence (AI) on employment and wages? Using administrative data on a comprehensive set of AI applications developed in Brazil, I show that AI is widely used not only in administrative tasks but also in the production setting, where it primarily supports process optimization and quality control. Exploiting exogenous variation in AI development costs, I find that AI affects administrative and production workers differently. In the office, AI automates tasks, decreases employment, and hollows out the wage distribution. In the factory, it increases employment of low-skilled and young workers by enabling them to perform complex tasks that previously required expertise. Overall, AI increases employment in Brazil because the gains in production outweigh the losses in office work.

Key Words: Artificial Intelligence, Automation, Deskilling, Software, Inequality

JEL: J23, J24, F63

* This paper benefited from valuable comments from Ajay Agrawal, Michael Andrews, David Autor, Raj Chetty, Oren Danieli, John Friedman, Nathaniel Hendren, Anders Humlum, Xavier Jaravel, Joe Kaboski, Aakash Kalyani, Kristina McElheran, Anna Salomons, and Yongseok Shin. I also thank seminar participants at the Opportunity Insights Conference on Economic Mobility, the Federal Reserve Bank of San Francisco, the Federal Reserve Bank of Chicago, IMF, LACEA, SBE, and Steg Annual Conference. Jack Mannion and Vitor Santos provided superb research assistance. I thank Haishi Li and Vitor Santos for providing the survey data.

† Federal Reserve Bank of Chicago. Contact: gustavo@microtomacro.net. This paper represents my opinions and not those of the Federal Reserve Bank of Chicago or the Federal Reserve System.

"I like to think (and
the sooner the better!)
of a cybernetic meadow
where mammals and computers
live together in mutually
programming harmony."

—Richard Brautigan,
"All Watched Over by Machines of Loving Grace"

1 Introduction

Artificial intelligence (AI) refers to computer systems that learn from data to recognize patterns, make predictions, or generate information—tasks that once required human labor. Advances in hardware and the widespread collection of digital data have made AI a quiet, everyday technology. It runs social media feeds, supports banking transfers, guides self-driving cars, automates industrial processes, and powers chatbots, among many other uses (Marr and Ward 2019). This breadth of applications has drawn public attention to AI's potential effects on the labor market. One view holds that AI automates tasks and displaces workers, especially in routine cognitive jobs (Acemoglu and Restrepo 2019, Webb 2020, Aum and Shin 2025). A contrasting perspective argues that AI can augment labor by simplifying tasks and improving decision-making, ultimately creating jobs (Autor 2024, Agrawal et al. 2019b, Bresnahan 2021). Despite growing interest, there is still no consensus on whether AI is, on net, decreasing or increasing employment.

A possible reason for these contrasting views on the effects of AI is that much of the existing literature studies AI in specific settings, reaching different conclusions. Some focus on generative AI tools used by professionals (Humlum and Vestergaard 2025b, Brynjolfsson et al. 2025), while others focus on the effects of AI in firms that develop their own technologies in-house (Hampole et al. 2025, Babina et al. 2024, Adams et al. 2024). Yet, evidence suggests that the scope of AI adoption is broader. AI is increasingly embedded in third-party software, reaching firms through off-the-shelf tools rather than internal development (Aoun et al. 2022, Xu et al. 2018, Li et al. 2024). Engineers highlight a wider set of applications, including factory settings, where AI appears to raise machine productivity and operational efficiency (McElheran et al. 2025, Alam et al. 2024, Marr and Ward 2019). This diversity of applications could lead to different conclusions because the effects of AI depend on how the technology is applied (Aghion et al. 2025, Ide and Talamàs 2025). Therefore, to understand

AI's overall labor-market impact, I need data that captures a wide and representative range of AI applications.

I contribute to this debate by studying how AI affects the labor market using a new administrative registry that covers a comprehensive set of commercial AI software developed in Brazil. The dataset offers detailed technical descriptions of each program and information about its intended use within firms, spanning from the earliest commercial AI applications through generative AI tools developed by 2023.[1] It captures both software built for internal company use and products developed by specialized IT firms and sold to other organizations.

Using exogenous variation in AI development costs as an instrument, I find that AI affects administrative and production workers in different ways. Among administrative workers, AI reduces employment and hollows out the wage distribution, mirroring earlier waves of automation (Autor et al. 2003, 2006, Autor and Dorn 2013, Goos et al. 2014, Michaels et al. 2014). In contrast, among production workers, AI increases employment of young, inexperienced, and low-education workers. In these jobs, AI raises wages for low-skilled workers but lowers them for high-skilled ones, consistent with experimental evidence (Brynjolfsson et al. 2025, Peng et al. 2023, Kanazawa et al. 2022). More generally, AI expands employment in occupations that use operational information to guide decisions and actions, such as production work, while replacing occupations centered on gathering and processing information, as theorized by Agrawal et al. (2019b). In aggregate, employment gains among production workers dominate, resulting in higher employment and lower inequality.

These contrasting effects help reconcile competing views on the labor-market impact of AI. In production, the results align with the idea that AI increases machine productivity, simplifies operation, and raises demand for low-skilled labor (Autor 2024, Bresnahan 2021, Agrawal et al. 2019b). In office settings, they are consistent with AI automating tasks and reducing employment (Acemoglu and Restrepo 2018, Webb 2020, Aum and Shin 2025, Hui et al. 2024). Taken together, the findings suggest that the impact of AI depends on how and where it is used, explaining why prior studies, focused on specific applications or firms, reached different conclusions.

To see these effects in action, consider Techplus, a Brazilian firm that develops AI systems for industrial operations. Its software, which is part of Brazil's software registry, analyzes real-time sensor data from factory equipment to predict failures, schedule maintenance, and

[1]The dataset includes generative AI applications developed up to 2023. However, most of the sample consists primarily of predictive AI software, which has been in use since the 2000s (McElheran et al. 2023, Brynjolfsson and McElheran 2019).

assist machine operation. In doing so, the software provides real-time information and expert knowledge to supervisors and machine operators. Such systems can reduce unplanned downtime by up to 50% (Agoro 2025, Benhanifia et al. 2025), which raises the value of keeping machines running and increases demand for workers who operate them (Schmitz 2005). At the same time, the software automates routine administrative tasks such as maintenance planning and inventory control, which is likely to reduce the demand for workers performing these information-processing tasks. This case reflects the broader pattern I document in the data: AI increases employment in low-skilled production roles by making expertise available, while reducing demand for routine office work through automation.

To study the effect of AI on the labor market, I begin by assembling a dataset on AI software developed in Brazil. In 1987, Brazil introduced copyright protection for the source code of computer programs registered with the National Institute of Industrial Property (NIIP). Subsequently, registering software became standard industry practice, with approximately 92% of software development firms in 2016 having at least one program registered.[2] Firms must report each program's title, intended use, and technological features, such as the use of AI methods like neural networks and deep learning. I collect data on all registered programs that use AI technology. Because NIIP records contain concrete technical and application details on actual deployed software, they offer a clearer measure of AI development than patents or job advertisements, which are the standard proxies in the existing literature.[3,4]

The NIIP registry shows four relevant facts about AI development. First, AI creation surged after 2013, tracking global advances in deep learning and graphics processing unit (GPU) computing. The annual filings increased from 90 in 2014 to 420 in 2022.[5] Second, AI is not confined to administrative tasks. While 48 percent of AI software targets office

[2]To assess the coverage of the registry, I also conducted a survey of industry specialists. Ninety-six percent of respondents are aware of the NIIP registration process or have taken part in it directly. Seventy-one percent report that most or all firms register at least one version of their software.

[3]Only 10 percent of firms with a software registration have a patent application, illustrating the broader coverage of the NIIP software registry.

[4]The main limitation of this dataset is that it does not cover software developed abroad. However, due to regulatory barriers and a strong home bias in computer services (Alaveras and Martens 2015, Miroudot et al. 2010, Harms and Shuvalova 2020, Ezell et al. 2013, Benz and Jaax 2022), foreign software plays a minor role in the Brazilian market. In 2015, imported computer services accounted for only 9.3% of domestic consumption, less than half the OECD average of 19%, and fewer than 0.02% of IT firms were foreign-owned. As a result, the majority of software used in Brazil is developed locally and thus captured by the NIIP registry.

[5]The boom in AI development in the 2010s has been documented in several prior studies (Toosi et al. 2021, Chauvet 2018, Sevilla et al. 2022). According to Kaplan et al. (2020), the AI boom was driven by an abundance of data, more powerful computers, and advances in machine learning.

or clerical functions, 44 percent is designed for core production activities in manufacturing, agriculture, construction, and other sectors. Third, production-oriented AI is concentrated in a few practical applications: 40 percent are in software to manage production, 21 percent handle automated quality inspection, and 15 percent are industrial robot interface software. Fourth, across both office and production settings, these AI applications primarily process data to forecast outcomes, classify states, or detect conditions, providing a continuous flow of information to management and production workers. The intensive use of AI in production challenges the idea that AI affects only white-collar work and points instead to substantial labor-market implications for machine operators and other shop-floor workers.[6]

To estimate how AI affects different occupations, I calculate an occupation-level AI exposure over time. The AI exposure is constructed using text similarity between the tasks performed by each occupation and the descriptions of AI software in the NIIP database, following standard approaches in the literature (Webb 2020, de Souza and Li 2023, Kogan et al. 2023, Hampole et al. 2025). The intuition is simple: occupations whose tasks resemble those performed by AI software are more exposed to AI development, which can either complement or substitute their work. By conducting the analysis at the occupation level, the AI exposure captures workers' exposure to both in-house and vendor-supplied AI and the resulting reallocation of workers across firms and sectors.[7]

To estimate the causal effect of AI, I use an instrumental variable strategy that exploits exogenous variation in the cost of developing AI software. The instrument relies on the fact that software development costs depend on the availability of resources for a programming language, such as libraries, tutorials, and community support, allowing developers to reuse scripts rather than build them from scratch (Delorey et al. 2007, Krein et al. 2010, Tambad et al. 2020, Vijayaraghavan et al. 2022). Therefore, an increase in the availability of resources for a programming language in the United States decreases the cost of developing AI software in Brazil. The key identifying assumption is that shocks to the popularity of programming

[6]Although the use of AI in production settings has received little attention from economists, it is widely studied in the engineering and operations literature. For example, Chen et al. (2006), Capgemini (2020), and Shojaeinasab et al. (2022) document the integration of AI into Manufacturing Execution Systems (MESs). AI is used in quality control for detecting anomalies and improving inspection accuracy, as discussed by Ghelani et al. (2024), Tariq et al. (2024), Elsafty et al. (2024), Shi et al. (2023), Yang et al. (2023), and Patel and Ghelani (2024), among many others. Additionally, AI is used to optimize energy usage in manufacturing (Evans and Gao 2016, Liu et al. 2022). Finally, Marr and Ward (2019) and Borboni et al. (2023) discuss the use of AI in the operation of industrial robots.

[7]I conduct the analysis at the occupation level because 81% of software developed in Brazil is produced by specialized IT firms and licensed to many clients; a firm-level analysis would therefore miss much AI adoption through third-party vendors.

languages in the US do not correlate with labor market shocks in Brazil, for which I provide supporting evidence.

I find that AI increases employment while drawing in workers with lower observable and unobservable skills. A one-standard-deviation increase in AI exposure raises occupational employment by 1.9 percent, particularly among younger, less-educated, less-experienced, and lower-ability individuals. This compositional shift lowers average wages by 2 percent. Holding worker composition fixed, however, a one-standard-deviation increase in AI exposure increases wages by 0.5%. These results offer a first indication that AI increases the productivity of low-skilled workers more than that of high-skilled workers, a pattern I explore further.[8]

AI reduces wage inequality within occupations by compressing wages at the top of the distribution. A one-standard-deviation increase in AI exposure lowers the within-occupation standard deviation of wages by 3%. This effect is driven by larger wage declines among high earners: AI exposure has no effect on the first decile of wages but decreases the 9th decile by about 3%. This pattern is consistent with experimental evidence showing that AI tools boost productivity more for lower-skill workers than for experts (Brynjolfsson et al. 2025, Peng et al. 2023, Kanazawa et al. 2022).

AI also reduces inequality across occupations. In occupations with higher initial wages and education levels, AI exposure lowers both average earnings and years of education more than in other occupations. These declines are again driven by an influx of lower-ability workers. These results are consistent with the idea that AI lowers skill barriers and opens access to roles once reserved for highly qualified individuals (Autor 2024).

These average effects, however, mask important heterogeneity between office and production workers. Among production workers, AI raises employment but lowers average wages by shifting hiring toward younger, less-educated, and less-experienced workers. Within this group, it raises wages in the bottom decile by 1% but reduces wages in the top decile by 4%. These effects are more pronounced among workers who operate machinery directly. A possible explanation for these findings is a reduced skill-premium: as AI makes information available, the value of expertise declines, allowing firms to substitute toward low-skill workers (Autor 2024).

Administrative occupations show a different pattern. For these workers, AI reduces

[8] According to the model by Autor and Thompson (2025), these results are consistent with AI reducing the expertise requirement of each occupation, bringing in more low-skilled workers and lowering average wages.

employment without significantly affecting average wages. However, this stability in averages masks a hollowing out of the wage distribution: AI decreases wages among administrative workers in the middle of the income distribution. These findings are consistent with the idea that AI automates tasks done by white-collar workers (Acemoglu and Restrepo 2018, Webb 2020, Aum and Shin 2025, Hui et al. 2024).

Why does AI automate the office and expand the factory? The key difference is how occupations interact with information. AI processes real-time data and converts it into information that guides decisions and actions. I show that occupations whose core tasks consist of processing, organizing, or recording information, such as office clerks or accountants, are replaced by AI. By contrast, occupations that use information to guide decisions or actions, such as machine operators or managers, are complemented by AI. In these roles, AI expands the set of workers who can perform them, leading to employment growth driven in part by an influx of low-skilled and less-experienced workers who gain access to expertise embedded in AI systems.

This paper contributes to the literature on the labor market effects of AI by introducing a new dataset and an instrumental variable strategy that identifies a novel channel: AI raises employment by boosting the productivity of less-experienced, lower-skilled workers. Much of the existing research emphasizes AI's labor-substituting role through task automation (Acemoglu and Restrepo 2018). Others argue that AI can augment workers and lead to job creation by increasing productivity (Agrawal et al. 2019b, Bresnahan 2021, Aghion et al. 2025). Agrawal et al. (2019b), Agrawal et al. (2019c), and Agrawal et al. (2019a) argue that AI provides a better prediction technology, which can increase demand for complementary tasks, such as judgment. Despite growing interest, empirical findings remain mixed. Some studies find little to no overall impact of AI on employment or wages, such as Humlum and Vestergaard (2025a) on AI chatbots, Acemoglu et al. (2022) using job postings, and Hampole et al. (2025) using resume data. Hampole et al. (2025), in particular, finds that the productivity gains are larger than the displacement effect at firms adopting AI. Others, including Webb (2020), Aum and Shin (2024), and Aum and Shin (2025), show that AI adoption can reduce earnings and job opportunities. Hosseini and Lichtinger (2025) finds AI to decrease the employment share of young workers, differently from standard software investment (Barth et al. 2023). In contrast, firm-level studies like Babina et al. (2024) and Adams et al. (2024) find positive effects of AI on firm growth and productivity, pointing to possible employment gains. Aghion et al. (2025), using firm-level data in France, finds AI to

increase employment but decrease it in case of administrative applications of it. Similarly, McElheran et al. (2025), Brynjolfsson and McElheran (2016) and Brynjolfsson et al. (2021a), studying the adoption of AI, data driven decision, and predictive technologies, find them to increase TFP and employment only when complementary investments are made, with losses of productivity in the short-run. Chalfin et al. (2016) shows practical examples of how the use of machine learning techniques to the hiring of workers can outperform standard hiring systems.

This paper helps reconcile the conflicting perspectives by showing that the labor market effects of AI vary across occupations. As Webb (2020) and Aum and Shin (2025) have argued, I find that AI reduces employment and wages among administrative workers, consistent with the displacement effects emphasized by Acemoglu and Restrepo (2018), Acemoglu et al. (2022), Hampole et al. (2025). At the same time, I find that AI increases employment in production occupations, supporting the view of Agrawal et al. (2019b) and Bresnahan (2021), who argue that AI can act as a productivity-enhancing tool that complements labor. Rather than offering a single answer, my results suggest that the aggregate effect of AI depends on the composition of the labor force and how AI is used.

This paper contributes to the literature on how automation and technological progress affect labor markets. Seminal work by Graetz and Michaels (2018) and Acemoglu and Restrepo (2020) shows that the diffusion of industrial robots reduced low-skilled employment and wages. Building on this evidence, recent research moves to the firm level and shows that robot adoption changes employment and productivity within firms, with effects that can cumulate into sizable aggregate outcomes (Hubmer and Restrepo 2021, Martinez 2021). These effects are not uniformly negative: some studies find that adoption raises productivity enough to expand scale and increase employment at adopting firms (Jaravel et al. 2020, Humlum 2019). Automation also extends beyond industrial robots to other forms of capital and production technology (Boustan et al. 2022, de Souza and Li 2023). This broader adoption helps explain job polarization across countries, consistent with routine-biased technological change (Goos et al. 2014, Danieli 2026). Although automation can displace labor in adopting activities, reallocation and demand spillovers can offset part of these losses in other sectors (Autor and Salomons 2018). Technological progress can also generate new tasks that counterbalance displacement over time and decrease wage inequality in the long run (Autor et al. 2024, Kalyani et al. 2025).

I make two contributions to this literature. First, I study a new wave of technologi-

cal progress and show that, unlike earlier waves of innovation, AI complements low-skilled workers and leads to lower inequality. Second, I broaden the measurement of automation technologies by using a dataset with detailed descriptions of software developed in Brazil. Most studies of automation focus on industrial robots (Humlum 2019, Jaravel et al. 2020), machines (Boustan et al. 2022, de Souza and Li 2023), or patents (Hémous et al. 2025). I show that AI software is also an automation technology for administrative workers.

The paper is organized as follows. Section 2 describes the data and the institutional context. Section 3 presents stylized facts about AI development in Brazil. Section 4 outlines the empirical strategy. Section 5 discusses the main results, and Section 6 addresses robustness. Section 7 concludes.

2 Data

This section introduces the data and explains why Brazil offers an unusually clear setting for studying the labor-market effects of AI.

The primary data source is the software registry maintained by the National Institute of Industrial Property (NIIP), which contains a broad sample of in-house, third-party, and commercially marketed software developed in Brazil(Barbosa et al. 2022). This registry is unusually comprehensive for two reasons. First, Brazil has a mature and self-sufficient IT sector: it is the 11th largest in the world and has the third-largest pool of software developers (GitHub Staff 2024). Second, regulatory barriers limit the import share of computer services to about 9%, meaning most software deployed by Brazilian firms is locally developed and therefore captured in the NIIP database.[9]

I use the NIIP registry to construct occupation-level measures of AI exposure. Then, I link this exposure measure to the RAIS administrative dataset, which tracks all formal jobs from 2003 to 2022 in narrowly defined 6-digit occupational codes, allowing precise measurement of employment and wage dynamics.

[9]I discuss in detail the import share of computer services in Appendix A.2.

2.1 Administrative Software Registry Data

2.1.1 Institutional Background

Brazilian software law provides special copyright protection for software. Since 1987, the Brazilian government has allowed companies to register their software with the National Institute of Industrial Property (NIIP) under a special copyright system that grants 50 years of copyright protection.[10] The registration process is similar to that of patents but less demanding: it requires no claims of novelty or inventive step, involves no technical review, charges low fees, and keeps the source code confidential. While it does not grant exclusive rights over the underlying idea, as patents do, it protects against copying or distribution of the software itself, offering firms effective protection against piracy. Piracy of protected software carries criminal penalties of up to four years in prison and substantial fines, giving firms strong incentives to register even routine applications.[11]

A software registration contains detailed information about the software. To register software, firms must submit an electronic form with the firm's name, the programmers' names, a descriptive title of the software, and the programming languages. Importantly, firms provide two classifications that describe the software's purpose and technical characteristics.

The first classification, called "application domain," identifies the sector or area where the software will be used. There are 225 specific application domains grouped into 35 broad categories. For example, *MapForest*, an AI program that forecasts weather and predicts land productivity, is classified under Agriculture. The software *iMachine* developed by Techplus, discussed in the introduction, has the application domains "Factory Planning", "Metrology", "Information Management", and "Manufacturing Technology". Each software entry includes at least one application domain, though most list multiple; on average, each program lists three. Table A4 in the Appendix lists all 35 broad application domains together with their descriptions.

The second classification, called "technical class," organizes software by its technical

[10] In Portuguese, the NIIP is called *Instituto Nacional de Propriedade Industrial*. The first software law was Law number 7,646/1987. It was later updated by Law 9,609/1998. The law defines software as a "set of instructions or statements, written in proper language, to be used directly or indirectly by a computer to obtain a certain result."

[11] The law aimed to strengthen intellectual property rights in the IT sector and encourage investment. It was part of a broader set of industrial policies targeting the sector in the late 1980s (Stuber 1984).

function. The system includes 97 specific technical classes divided into 18 broad groups, ranging from fundamental systems like Operating Systems and Network Controllers to specialized tools like Encryption Software and AI. *MapForest*, for instance, is classified under four technical classes: Artificial Intelligence, Simulation, Pattern Recognition, and Office Automation. In the empirical analysis, I focus on software with technical class Artificial Intelligence. Table A3, in Appendix A.2.2, lists the broad technical classification groups with their descriptions.

Software registration provides proof of ownership. Once software is registered, the firm receives a registration certificate that serves as official legal proof of ownership. Figure 1 shows one such certificate. These certificates are commonly used in cease-and-desist letters, piracy takedown requests, and intellectual property lawsuits.[12]

Why are firms registering their software? Both firms and developers have strong incentives to register software, making the NIIP database a comprehensive record of software development in Brazil. Specialists typically list five reasons firms register their software: intellectual protection, government procurement, access to credit, signaling to investors, and the low financial cost (Torquato 2022). First, registration provides legal proof of ownership and forms the basis for asserting rights against unauthorized use. Second, registration is mandatory for participating in government procurement of software or IT services. Third, firms seeking R&D credit must show proof of software registration as evidence of prior innovation. Fourth, registration signals the value of a firm's intangible assets to investors. According to the Brazilian Association of Software Firms, most startup investors require all software developed by a firm to be registered (Associação Brasileira das Empresas de Software (ABES) 2024). Finally, the process is inexpensive, with the NIIP charging just $34 per registration.

Developers also benefit from software registration. Developers have equally strong incentives. Each registration lists the programmers involved in the project, providing formal proof of experience that developers include on résumés and share on platforms like

[12]In the registration process, firms also provide a hash summary: a unique, fixed-length string generated by applying a cryptographic hash function to the source-code. Because any modification to the code would alter the hash, this digital "fingerprint" serves as proof that the firm possessed that exact version of the code on the filing date. The hash can later be compared to the confidential source code to establish authorship in legal proceedings.

Figure 1: **Example of a Software Registration**



FEDERATIVE REPUBLIC OF BRAZIL
MINISTRY OF THE ECONOMY
NATIONAL INSTITUTE OF INDUSTRIAL PROPERTY
DIRECTORATE OF PATENTS, COMPUTER PROGRAMS, AND TOPOGRAPHIES OF INTEGRATED CIRCUITS

Certificate of Software Registration

Case Nº: **BR512021001026-4**

The National Institute of Industrial Property issues this certificate of registration of a computer program, valid for 50 years starting from January 1st following the date of May 3, 2021, in accordance with §2, Article 2 of Law No. 9,609 of February 19, 1998.

Title: iMachine AI: Online Predictive Maintenance System for Electric Machines Using Machine Learning Techniques

Publication Date: 05/03/2021

Creation Date: 04/28/2021

Owner(s): TECHPLUS AUTOMACAO

Author(s): SAMARONE RUAS

Programming Language: PYTHON

Application Domain: AD-06; FQ-04; IF-10; IN-02

Technical Class: AP-01; AT-05; IA-01; IT-01; TC-03

Hash Algorithm: SHA-512

Digital Hash Summary:
cec9fe6467e2da2a9cc51fad00cfaedf52adc26dd5f352d4f4ed7c8600e399415383cc
2d542eab52cf2a49bf66955b20317e
bf21bafa7273d7edcdb534767588

Issued on: 05/25/2021

Approved by:
Carlos Alexandre
Fernandes Silva Head
of DIPTO

Notes: This figure shows a translated certificate of software registration. The application domains correspond to "factory planning", "metrology", "information management", and "manufacturing technology". The technical classes are "application", "industrial automation", "artificial intelligence", "instrumentation", and "pattern recognition".

LinkedIn.[13] This credentialing carries substantial economic value: de Souza et al. (2024) documents that programmers receive a 5% permanent wage increase after being listed on a registered software project. Moreover, in Brazil's public-sector hiring for academics or IT professionals, applicants with a software registration are explicitly ranked above those without.[14] Therefore, because registration directly benefits their careers, it's reasonable to think that developers pressure employers to register the software they build. Combined with the firm-level incentives described above, this bottom-up pressure likely contributes to the high registration rates documented in the coverage statistics and survey evidence below, supporting the view that the NIIP database captures a broad sample of commercially developed software in Brazil.

[13]Figure A1 in the Appendix shows two examples of developers displaying their certificates.
[14]IFPE (2025) is one out of many such cases.

2.1.2 Administrative Software Registry Data

I extracted information from the universe of NIIP certificates of software registration between 1987 and 2023. For each entry, I observe all the information in the certificate of software registration (Figure 1): the software title, the publication and creation dates, the names of the owners, the programmers involved, the programming languages used, the application domains, and the technical classifications. In the empirical section, I focus on AI software, which are those with technical class "Artificial Intelligence"("IA01").

There is a total of 37,677 registrations, which, importantly, does not reflect the number of distinct software products. Registration is a legal tool, not a technical unit. A comparison of the software registrations with public online records reveals that firms usually bundle different software products in one registration. Legally, this approach offers the same level of protection as if the firm were to register each program separately. For the purposes of this paper, what matters is that commercially relevant software is typically registered at least once with a description of its intended use, even when firms bundle multiple products within a filing. I discuss the nuances of this database in further detail in the next section.

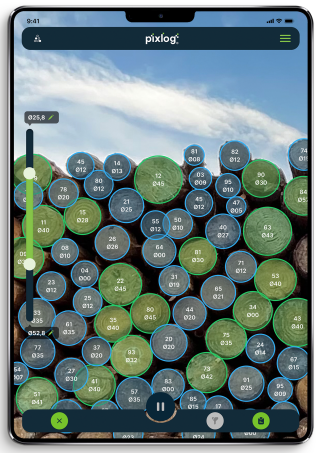
2.1.3 Selected Examples

To illustrate the results that follow, it is useful to consider a concrete example from the database. Figure 2 shows Pixlog, an AI-based mobile application and management platform that automates log measurement and verifies whether shipments meet contractual specifications. Traditionally, loggers measure the diameter and length of each log by hand and repeatedly compute volumes across large timber piles. These tasks are hard to perform accurately in the forested and remote areas where logging takes place. They are especially difficult for less-educated workers, who commonly work in these locations, increasing the risk of mismatches between customer orders and delivered products (Mais Floresta 2023). By embedding measurement expertise in the tool and providing accurate operational information, Pixlog enables production workers to complete these tasks faster and more consistently, raising their productivity. At the same time, the platform automates administrative functions such as inventory tracking and order control, reducing the need for manual recordkeeping.

Another example from the database is the generative AI software developed by Hyper Reports. It provides an analytics automation platform that uses generative AI to collect information from multiple company systems, including sales, inventories, and machine operation logs, and organize it into real-time reports. Workers can easily access operational

information from the company through a WhatsApp chatbot. As a result, the software broadens access to information while automating the tasks of compiling, formatting, and distributing reports.

Figure 2: **Example of AI from the Database**



Notes: Image from Pixlog (n.d.). Pixlog is a software registered in the NIIP.

2.1.4 Coverage, strengths, and limitations

This subsection discusses the NIIP registry's coverage, advantages, and limitations. In terms of coverage, registration is common industry practice: 92% of software development firms with more than two years of operation have a software registration, a finding validated by a survey of industry specialists. The NIIP's key advantage compared to other databases used to measure AI adoption, such as patents or job advertisements, is that it captures actual software deployed by firms, including AI tools that lack patent-worthy novelty and third-party vendor software missed by hiring data. However, the data have limitations related to scope and registration practices, but as I discuss below, these are unlikely to affect the results.

Registration is nearly universal among software developers. In 2016, 92% of software development firms with more than two years of operation had registered at least one

software program.[15,16] When weighted by employment, this figure rises to 99.4% of software development companies. Among larger developers, registration is nearly universal: 96% of firms with more than 20 employees have at least one program registered with the NIIP.[17] These statistics indicate that registration is widespread among Brazilian software developers, consistent with the NIIP registry capturing a large share of commercially developed software created in Brazil.

Survey with private sector specialists. To validate the dataset and better understand the degree of selection, I conducted a survey with IT firms and intellectual property lawyers, described in more detail in Section A.2.1. The survey provides three main findings. First, software registration in the NIIP is widely known among both intellectual property lawyers and IT professionals, with the vast majority of respondents having direct experience with the registration process. Second, respondents confirm that most commercial software developed in Brazil has at least one version registered, with firms primarily motivated by intellectual property protection. Third, to the extent that selection exists, it is driven by small firms producing consumer-facing applications such as video games and mobile apps. These categories are unlikely to affect the labor market outcomes studied in this paper.

Comparison to other datasets. The NIIP registry expands traditional metrics to measure AI adoption, such as patents or job ads. Unlike patents, which require novelty and an inventive step, NIIP accepts any original code. This lower threshold allows it to capture routine but economically important AI tools that would not qualify for patent protection. The data support this conclusion: only 10% of firms with a software registration also hold a software patent. Furthermore, while job advertisements primarily capture in-house development, often missing AI developments from specialized IT companies, the NIIP registry includes registrations from specialized IT vendors, capturing off-the-shelf software tools that are widely adopted but not built in-house. Finally, the registry provides richer technical

[15]I call a firm a software development firm if it has Classificação Nacional de Atividades Econômicas (CNAE) 2002 codes 6201500 (custom software development) or 6202300 (development and licensing of customizable software), and has hired a computer engineer at least once in its life-cycle.

[16]I restrict the sample to firms with more than two years of operation because, as shown in Section B.4, software development typically requires more than two years.

[17]This figure represents a conservative lower bound on overall coverage because not all of these firms have developed their software: (i) some firms might only provide technical support and do not write proprietary code, (ii) some firms are third-party developers whose software is registered by the client, and (iii) young firms are still completing their first product. As shown in Section B.4, software development takes more than two years for 50% of firms.

detail than typical hiring data; each entry includes the software's title, technical features, and intended application domain, allowing for a precise link between specific AI technologies and the occupations they affect.

Weaknesses. The NIIP database has three main limitations: it covers only domestically developed software, treats registrations as legal rather than technical units, and typically records only a program's first version. Despite these limitations, to my knowledge, the NIIP database remains the only dataset with information on actual software deployed by firms, making it uniquely valuable for studying AI's labor market effects.

The registry covers only software created in Brazil, omitting software adopted locally but developed abroad. In practice, however, this omission is quantitatively small. Imported computer services, which include software publishing, programming, data processing, and cross-border license fees, accounted for only 9% of domestic computer service consumption in 2015—less than half the OECD average of 19%—and fewer than 0.02% of IT firms were foreign-owned in 2016.[18]

This low import share reflects two key factors. First, regulatory barriers—including data-localization requirements and domestic-content rules—raise the effective trade cost to the equivalent of an 87% tariff (Ezell et al. 2013, Benz and Jaax 2022). Second, digital services face inherently high trade costs globally, typically two to three times higher than for physical goods, due to cultural and regulatory frictions that create a strong home bias even in open economies (Miroudot et al. 2010, Harms and Shuvalova 2020, Alaveras and Martens 2015). I discuss these factors in more detail in Section A.2. Given that foreign software accounts for less than one-tenth of the market, the NIIP registry effectively approximates the software landscape that Brazilian workers face.

A second limitation arises from the legal nature of software registrations. Because a registration protects the source code submitted—regardless of how many distinct programs it contains—firms often bundle multiple software programs into a single filing. This means the total count of registrations (37,677) understates the actual number of individual programs. To address this, the empirical analysis relies on the textual content of each registration rather than on the count of filings. Even when firms bundle several programs together, the descriptions of each program's technical features and intended applications are preserved in the registry, allowing me to identify which AI technologies are deployed and which occupations

[18]Employment-weighted, about 0.03% of IT firms and 0.04% of IT and telecom firms were foreign-owned.

they affect.

Finally, firms often register only the first version of a program and may not file updates, meaning late additions of AI features could go unrecorded. This would attenuate the observed relationship between AI adoption and labor-market outcomes, suggesting that my estimates reflect a conservative lower bound on the true impact of AI technologies.

2.2 Other Datasets

I also use four additional datasets. Labor market data comes from RAIS, and task descriptions from the Brazilian Ministry of Labor. To construct the instrument, I use data from Stack Overflow posts and information on the popularity of programming languages in the U.S., drawn from TIOBE.

RAIS. I use the publicly available version of *"Relação Anual de Informações Sociais"* (RAIS), a matched employer-employee dataset covering all formal workers in Brazil. It reports each worker's occupation, earnings, hours, and other job details. I rely on this version because it extends coverage through 2022.[19] For privacy reasons, the public version excludes person identifiers, so I use a panel at the occupation–year level. Since 2003, RAIS has used the occupational classification system *Classificação Brasileira de Ocupações 2002* (CBO 2002), which includes detailed occupation descriptions. To avoid crosswalk issues with older classifications, I restrict the analysis to 2003–2022. I exclude from the sample IT workers, government employees, and military personnel.

Description of Occupational Tasks. To measure workers' exposure to AI, I use data from the Ministry of Labor, which links each six-digit occupation in the CBO 2002 classification to detailed descriptions of its tasks. Because the CBO is used in labor-court proceedings to adjudicate responsibilities and regulatory compliance, these descriptions are long-form, legally precise, and highly specific. The dataset covers 2,641 occupations, each with an average of 64 task statements.

Popularity of Programming Languages. I construct an instrument using historical trends in U.S. programming language popularity from TIOBE, which tracks the number

[19]The longitudinal version tracks workers over time but only runs through 2016, missing important years of AI development.

of search results on google.com for each programming language.[20] This approximates the number of English-language webpages hosted in the United States that mention each programming language, serving as a proxy for the size of the U.S. developer community, available resources, and libraries for each language. Because these trends reflect the evolution of the U.S. software industry rather than labor-market conditions in Brazil, they provide a source of variation plausibly exogenous to Brazilian employment outcomes. The data covers the period from 2000 to 2023.

Stack Overflow. To link programming languages to NIIP's technical classifications, I use data from Stack Overflow, a widely used question-and-answer platform for programmers. On this platform, programmers post questions seeking help from the community on specific technical problems. Each post includes standardized tags indicating the programming language and type of application involved, selected from a community-maintained list. Because tags often reflect both the programming language and the technical area of the question, they reveal how languages map to software technical classes. I collect data on all Stack Overflow posts created between 2008 and 2022.[21]

3 Statistics of AI Software

This section presents four key facts about AI software development in Brazil; additional statistics are reported in Appendix A.3. First, following global breakthroughs in AI development, the number of firms in Brazil that own AI software increased more than tenfold between 2010 and 2023. Second, AI is applied across a wide range of fields. While administrative uses such as report generation remain common, 40% of application-domain classifications for AI software target production settings, including manufacturing, agriculture, and logistics. Third, two applications dominate production use: manufacturing execution systems for workflow management and quality-control tools that perform real-time visual inspections. Fourth, across both office and production settings, AI applications primarily process data to forecast outcomes, classify states, or detect conditions. Therefore, AI is primarily used as a prediction technology (Agrawal et al. 2019b, 2018).

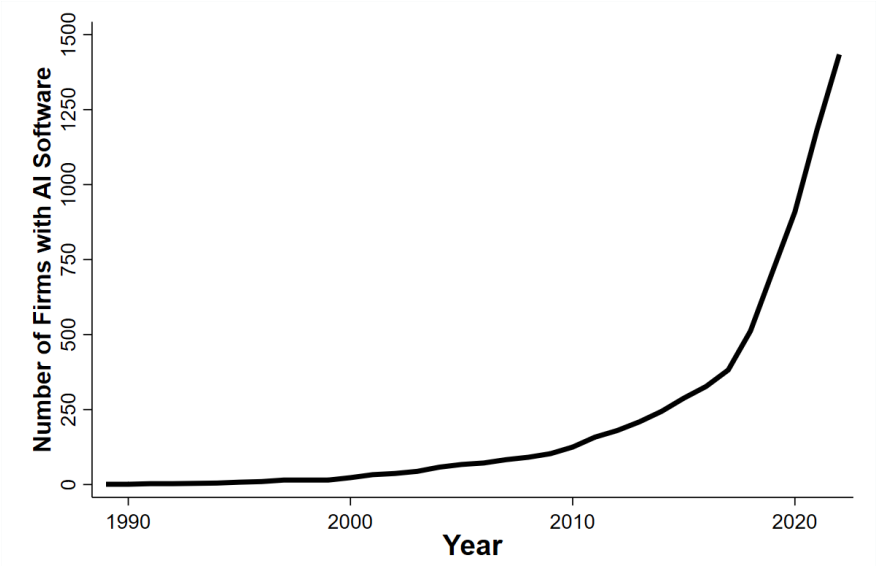
[20] For more information, see `https://www.tiobe.com/tiobe-index/`.

[21] Figure A17 plots the most common tags, which often reflect both the programming language and the technical area of the question.

Surge in the creation of AI software following major technological breakthroughs. Figure 3 plots the number of unique firms with at least one AI software registration.[22] Because 81% of these firms develop software for external clients, the number of firms with AI software registrations should be interpreted as a lower bound on AI adoption.[23]

Figure 3 shows a marked increase in the number of firms with AI software starting in the 2010s. From 2013 to 2022, the number of firms owning AI technologies increased sevenfold, reaching 1,434 by 2022. This period, known as the AI boom (Toosi et al. 2021, Chauvet 2018, Sevilla et al. 2022), was marked by a surge in AI development worldwide, fueled by scientific breakthroughs in neural networks and GPU computing (Metz 2021). It also saw a large increase in job postings for AI scientists (Acemoglu et al. 2022), AI patent applications (Webb 2020), and academic publications in AI (Maslej et al. 2025). Figure A12 in the Appendix shows that the rise in Brazilian AI software development followed the broader global surge, but at a slower pace.

Figure 3: **Number of Firms Owning AI Software**



Notes: This figure shows the number of unique firms in Brazil with at least one AI software registration from 1987 to 2023. Because many of these firms provide services to others, the number of firms using AI is likely much higher than those directly owning the software.

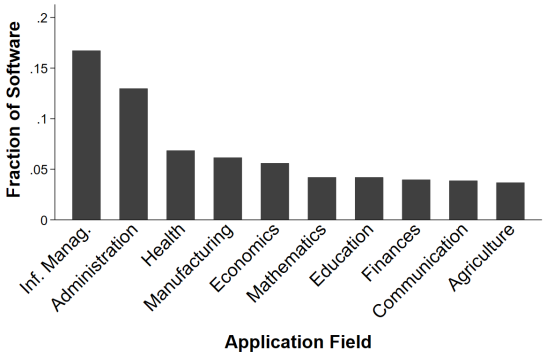
[22] I plot the number of firms rather than the raw count of registrations because, as discussed earlier, a registration does not reflect the number of distinct software products. In Appendix A.3, I show that the results are similar when using the share or number of AI software registrations.

[23] 81% is the share of firms with AI software in sectors related to software as service, including IT services, office support, and consulting.

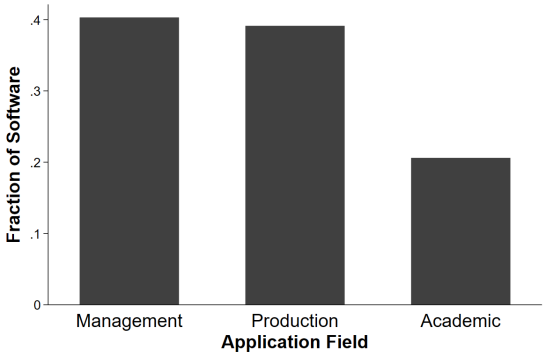
AI is used in management and production. When firms register their software, they indicate its application domains: a list of areas where the software is intended to be used. Figure 4a presents the ten most common broad application domains for AI software. Managerial domains, such as information management and administration, are the most frequent but account for only 27% of application-domain classifications. AI is also widely applied in fields like healthcare, manufacturing, and agriculture, which together represent 16.7% of the total.[24]

Figure 4: **AI Software Registrations by Intended Use**

(a) Top 10 AI Application Domains



(b) AI on Management and Production



Notes: These figures present the distribution of AI software across different application domains, as defined by the NIIP. The application domain classification was developed by the NIIP to indicate the areas in which each software is intended to be used. Because a single software program may be assigned to multiple domains, each figure reports the fraction of each domain relative to the total number of domain classifications. Specifically, Figure 4a shows, for each domain, the ratio of the number of times that domain appears across all software to the total number of domain labels used. Figure 4b groups these domains into three broad categories. The "Management" category includes domains related to business administration, law, economics, finance, information storage, telecommunications, and human resources. The "Production" category includes agriculture, civil construction, ecology, education, energy, physics, chemistry, geography, geology, housing policy, hydrology, engineering, meteorology, soil studies, sanitation services, public infrastructure, transportation, biology, botany, and health. The "Academic" category covers domains in mathematics, cultural studies, politics, public policy, psychology, environmental policy, social studies, city planning, and social welfare.

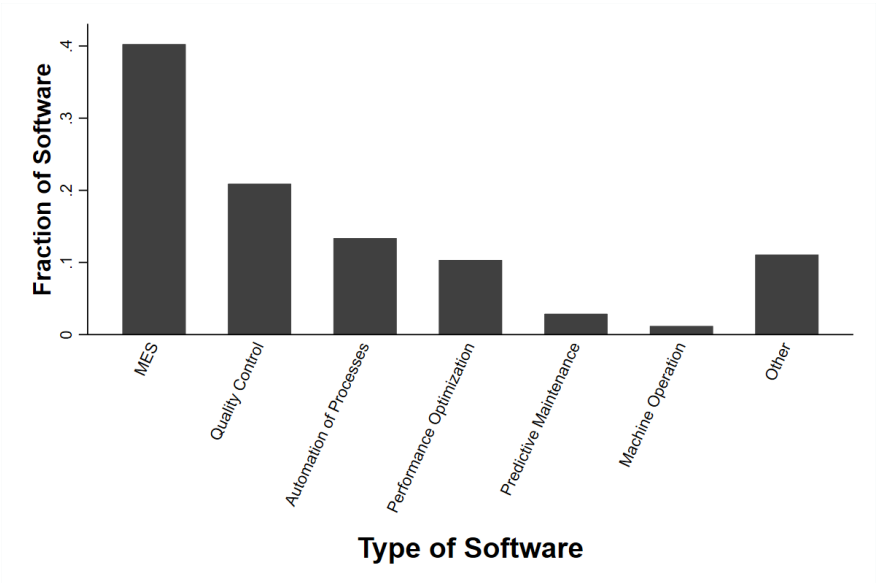
Figure 4b groups all application domains into three categories: management, production, and academic. Management includes back-office functions such as human resources, accounting, and administration. Production covers applications linked to the non-management applied domains, such as manufacturing, agriculture, and construction. The academic category includes research-oriented applications in fields such as mathematics and the social sciences.[25]

[24]Table A4 lists all application domains with descriptions.

[25]AI software with application domains in business management, law, economics, finance, information storage, telecommunications, or human resources falls into the "Management" category. AI with application domains in agriculture, civil construction, ecology, education, energy, physics & chemistry, geography, geology, housing policy, hydrology, engineering, meteorology, soil studies, sanitation services, public infrastructure, transportation, biology, botany, and health is categorized as "Production." AI used in mathematics, cultural studies, politics, public policy, psychology, environmental policy, social studies, city planning, and social welfare is classified as "Academic."

Figure 4b shows that AI is nearly evenly distributed between management and production applications, each accounting for about 40% of application domains. This distributional balance challenges the prevailing view that AI primarily targets high-skill, white-collar occupations (Frey and Osborne 2017, Webb 2020, Felten et al. 2021). With a comparable share of AI tools applied in production, blue-collar workers may be just as exposed to AI as administrative workers.

Figure 5: **Specific Use of AI Software in Production**



Notes: This figure shows the distribution of AI production software across application categories. Each software entry was classified into one of the groups based on its title, technological class, and application domain by a LLM.

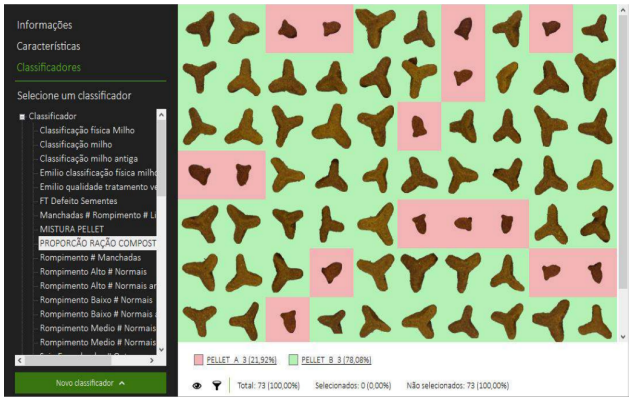
Production AI coordinates workflow, controls quality, and automates tasks. To better understand how AI is used in production, I classify each registration with a production-related application domain into one of seven mutually exclusive categories, based on the engineering literature and a manual review of program descriptions. Figure 5 plots the distribution of production AI software across the different categories.[26]

[26]To construct the AI classifications, I began by reviewing the engineering and operations literature on the practical applications of AI in production. This review identified several recurring use cases, including Manufacturing Execution Systems (MES) (Chen et al. 2006, Capgemini 2020, Shojaeinasab et al. 2022), quality control (Ghelani et al. 2024, Tariq et al. 2024, Elsafty et al. 2024, Shi et al. 2023), industrial automation through collaborative robots and robotic arms (Borboni et al. 2023, Marr and Ward 2019), predictive maintenance (Keleko et al. 2022, Ucar et al. 2024, Murtaza et al. 2024), performance-optimization software that reduces electricity use or other input consumption (Evans and Gao 2016, Liu et al. 2022), and machine operation software, such as the vision-based picking aids used in Amazon warehouses (Marr and Ward 2019). Finally, I added a seventh category for software not well suited to any of the above categories. To classify each software entry, I used a large language model (LLM) with frontier prompting practices: I provided

The most common use of AI software is in Manufacturing Execution Systems (MES), which coordinate and control production processes. MES accounts for nearly 40% of all AI production applications in the database (Figure 5). A concrete example is the AI software developed by Copel, a large electricity distribution company.

Copel's system monitors household-level energy usage and uses AI to predict power outages.[27] When risks are detected, the software automatically dispatches maintenance and inspection teams with a description of the issue, the repairs needed to be made, and the location, increasing the productivity of field workers. According to Consoli (2025), the software has prevented 1,700 outages. More broadly, AI-enabled MES has been shown to reduce production time, increase yields (Chen et al. 2006), and improve machine utilization rates (Capgemini 2020), which could increase productivity and the demand for workers.[28]

Figure 6: **Example of AI Software Used in Quality Control**



Notes: Example of AI software from Tbit. Image from Tbit (2024).

The second most common use of AI is in quality control (Figure 5), where software with computer vision detects defects in real time. A notable example in the database is Tbit's image-recognition system, which identifies visual impurities in products such as seeds, flour, and animal feed. In one deployment at a dog food factory, the software flagged defective units for removal from the production line, as illustrated in Figure 6. It also relayed defect rates

the Gemini with a description of each category (Kumar et al. 2023, Hansen et al. 2024, Ruckdeschel 2025), hand-coded examples for guidance (Brown et al. 2020), and decision rules (Halterman and Keith 2025, Fonseca and Cohen 2024, Shinohara and Kawazoe 2026), then instructed it to assign each AI software record to one of the predefined groups, providing it with all the information available in each software registration.

[27]The use of AI in electricity distribution is widespread. Eight other utility companies in Brazil have developed their own AI tools. For a review of how AI is used in electricity distribution, see Forootan et al. (2022).

[28]See Shojaeinasab et al. (2022) for a review of AI in MES.

and quality metrics to the Manufacturing Execution System (MES), which used the data to recalibrate machines, adjust input mixes, and guide operator decisions. As documented in the engineering literature, this integration of AI into quality control improves precision and productivity by reducing inspection time, minimizing human error, and lowering defect rates (Ghelani et al. 2024, Tariq et al. 2024, Shi et al. 2023).

AI is a prediction technology. Figure 7 shows a word cloud built from the titles of AI software. Aside from the prominence of the words artificial and intelligence, the remaining terms give a picture of how AI is used. Many of the most common words describe prediction tasks, such as forecast, recommendation, classification, recognition, and detection. Others point to areas where these predictions are applied, including facial analysis, documents, banks, debts, and financing. Words like machine, neural, deep, and learning describe the methods behind these tools. Essentially, the word cloud is consistent with viewing AI as a prediction technology (Agrawal et al. 2019b).

Figure 7: **Word Cloud from AI Software Titles**



Notes: This figure shows a word cloud from the titles of AI software.

4 Empirics

This section presents the empirical strategy used to estimate the effect of AI on the labor market. Following Kogan et al. (2023), de Souza and Li (2023), and Webb (2020), I first construct an occupation-level measure of AI exposure by computing the cosine similarity between task descriptions and AI software descriptions. To address differences in terminology, I apply latent semantic analysis, which maps synonyms and technical jargon onto shared se-

mantic dimensions. I then use an instrumental variable strategy that exploits heterogeneous exposure to changes in the popularity of U.S. programming languages. These changes affect the availability of AI programmers and libraries in Brazil, generating exogenous variation in AI development costs that is plausibly unrelated to labor-market conditions in Brazil.

4.1 AI Exposure

Text similarity between occupations and AI software. To measure how exposed each occupation is to AI, I link the tasks performed by workers to those carried out by AI software. This measure captures the overlap between AI software and workers' tasks, which could arise from either potential substitution or complementarity between them. I construct this measure in three steps. First, I collect detailed task descriptions for each occupation from the Brazilian Ministry of Labor. Occupations are defined narrowly at the six-digit level, giving 2,641 distinct occupations in total.[29] On average, each occupation has 64 tasks, for a total of 49,660 unique tasks. Second, I construct descriptions for each AI software record by combining its title and the description of the listed application domains. Third, I calculate the cosine text similarity, $\rho_{o,s}$, between all the tasks performed by each occupation o and the description of AI software s using latent semantic analysis. Appendix B.1 provides further details on the calculation of the cosine similarity.

Table 1 presents three illustrative AI programs and, for each program, the three occupations with the highest cosine similarity scores. The first program, whose software registration is shown in Figure 1, is a predictive maintenance tool most closely associated with occupations involved in machine maintenance and calibration. The second software automates the processing and sale of insurance policies and is strongly linked to insurance sales representatives and brokers. The third program analyzes soil samples to identify mineral content and is most similar to occupations such as miners and oil well cementers, whose tasks require identifying and studying minerals found in the soil.[30]

[29] As an example of the narrowness of the occupational classification, economists in agriculture, finance, manufacturing, public finance, environmental, and regional economics each have their own entry.

[30] Among the tasks performed by an oil well cementer are "prospecting, exploring, and analyzing mineral and soil data."

Exposure to AI. Using the cosine text similarity, I calculate the exposure of each occupation o to the AI software created before year t as:

$$AI\ Exposure_{o,t} = \sum_{s} \rho_{o,s} \mathbb{I}_s \left\{ t(s) \leq t \right\}, \tag{1}$$

where $\rho_{o,s}$ represents the text similarity between occupation o and software s, and $\mathbb{I}_s \left\{ t(s) \leq t \right\}$ is an indicator variable that equals one if software s, created in year $t(s)$, was developed in or before year t. Because firms register software right before commercialization, $t(s)$ is also the year when the software hits the market. Thus, $AI\ Exposure_{o,t}$ measures the cumulative exposure of occupation o to all AI software developed and commercialized up to year t. In Appendices B.2 and B.3, I show summary statistics on AI exposure.

Table 1: **Top Occupation Matches for Selected Software Titles**

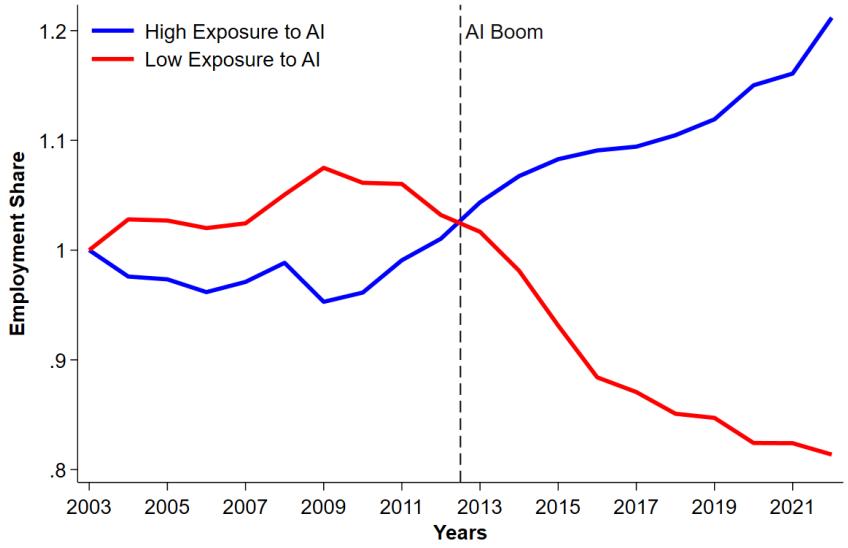
Occ. Code	Occupation Title	Similarity
Imachine AI: Online Predictive Maintenance System for Electric Machines Using Machine Learning Techniques		
3134-10	Instrumentation technician	0.265
3134-05	Calibration technician	0.265
3134-15	Supervisor of maintenance for control, measurement, and related instruments	0.262
Software for Insurance Sales and Customer Service Using Blockchain and Smart Contracts		
3517-40	Insurance technician	0.417
3545-05	Insurance broker	0.399
3517-05	Insurance analyst	0.389
Software with a Computational Model Based on Semi-Supervised Learning for Automatic Recognition of Potential Mineral Bodies		
3163-40	Oil well cementer	0.271
3163-35	Wellbore scale removal technician	0.280
7114-05	Miner	0.344

Notes: This table shows the occupations with the highest text similarity to each AI software record. Column 1 contains the occupation code, column 2 the occupation title, and column 3 the text similarity. Each panel shows the occupations with the highest cosine similarity for each AI software record.

Employment has increased faster in occupations more exposed to AI. Figure 8 plots the employment share of occupations in the top and bottom deciles of the 2022 AI-exposure distribution, normalizing both series to 1 in 2003. From 2003 to 2012, the two

employment trajectories moved together closely. Starting in the first year of the AI boom (2013), when the development of AI software exploded in Brazil and worldwide, employment in high-exposure occupations began to rise relative to low-exposure ones. By 2022, employment share in the most exposed occupations increased by 20%. Consistent with this pattern, Figure 9 shows a strong positive correlation between changes in AI exposure and employment growth from 2003 to 2022.

Figure 8: **Occupations More Exposed to AI Had Faster Employment Growth**



Notes: Figure 8 plots employment share for occupations in the top and bottom deciles of the 2022 AI-exposure distribution, normalizing both series to 1 in 2003.

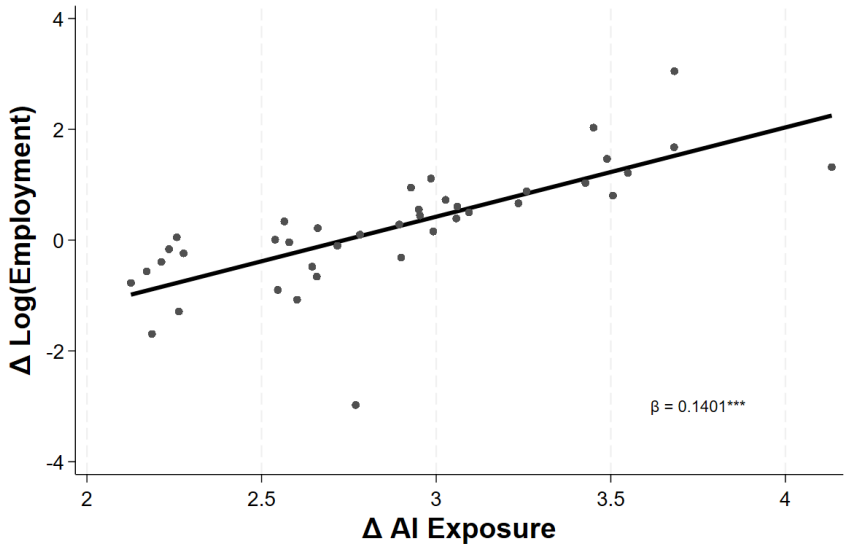
4.2 Empirical Model

Main empirical model. To estimate the effect of AI, I use the following empirical model:

$$y_{o,t} = \beta AI\ Exposure_{o,t} + \mu_o + \mu_t + X'_{o,t}\theta + \epsilon_{o,t}, \tag{2}$$

where $y_{o,t}$ is a labor market outcome for occupation o in year t, such as average wages or employment. $AI\ Exposure_{o,t}$ is the AI exposure measure defined in equation (1). Since the exposure measure is normalized to have mean zero and a standard deviation of one, β represents the effect of a one-standard-deviation increase in AI exposure on labor market outcomes for occupation o in year t. The terms μ_o and μ_t denote occupation and year fixed

Figure 9: **Exposure to AI is Correlated with Larger Employment Growth**



Notes: This figure shows the correlation between changes in AI exposure and employment growth from 2003 to 2022. The data are binned into 40 percentile bins according to the change in AI exposure. Then, I average the change in employment and AI exposure within each percentile bin. Each dot represents the average within each bin. The regression line is estimated using the raw observations and not the bins.

effects, respectively. Standard errors are clustered at the occupation level.[31]

Controls to deal with omitted variable bias. To limit omitted-variable bias, the control vector $X_{o,t}$ includes three terms designed to address specific confounders. First, I add a one-digit occupation-by-year fixed effect to absorb shocks common to broad occupational groups. Second, to account for exposure to foreign-developed software, I include a U.S. patent-based AI exposure measure, even though imported software accounts for less than 10% of the market, as discussed in Section A.2.[32] Finally, I control for potential trends by including the lagged outcome variable, $y_{o,t-1}$. As shown in the robustness checks, results remain consistent without these controls.

[31]Since each occupation o corresponds to a narrowly defined six-digit occupational code that often embeds sectoral requirements, I do not use a sector–occupation specification as in Webb (2020). In the robustness, however, I show that adopting a specification similar to that in Webb (2020) delivers qualitatively similar results.

[32]To construct exposure to U.S. patents, I follow the method proposed by Webb (2020).

Dynamic model. To test for parallel trends and capture the dynamic effects of AI on the labor market, I estimate the following model:

$$y_{o,t+j} = \beta_j AI\ Exposure_{o,t} + \mu_o + \mu_t + X'_{o,t}\theta + \epsilon_{o,t} \tag{3}$$

where β_j captures the effect of a one-standard-deviation increase in AI exposure on the labor market outcome $y_{o,t+j}$, measured j years after the software enters the market. I estimate (3) for horizons $j \in -3, 3$, where $j < 0$ are lagged outcomes used to assess pre-trends.

4.3 Instrument

Endogeneity concerns: trends and directed technological change. An OLS estimate of β in equation (2) may be biased for two reasons. First, software companies may direct AI development toward tasks performed by high-wage workers, generating a spurious positive correlation between AI exposure and wages. Second, AI adoption may coincide with other occupation-specific trends, confounding its true effect. To isolate the causal impact of AI, I instrument for AI exposure using variation in the cost of developing AI software, proxied by changes in the relative popularity of programming languages in the United States.

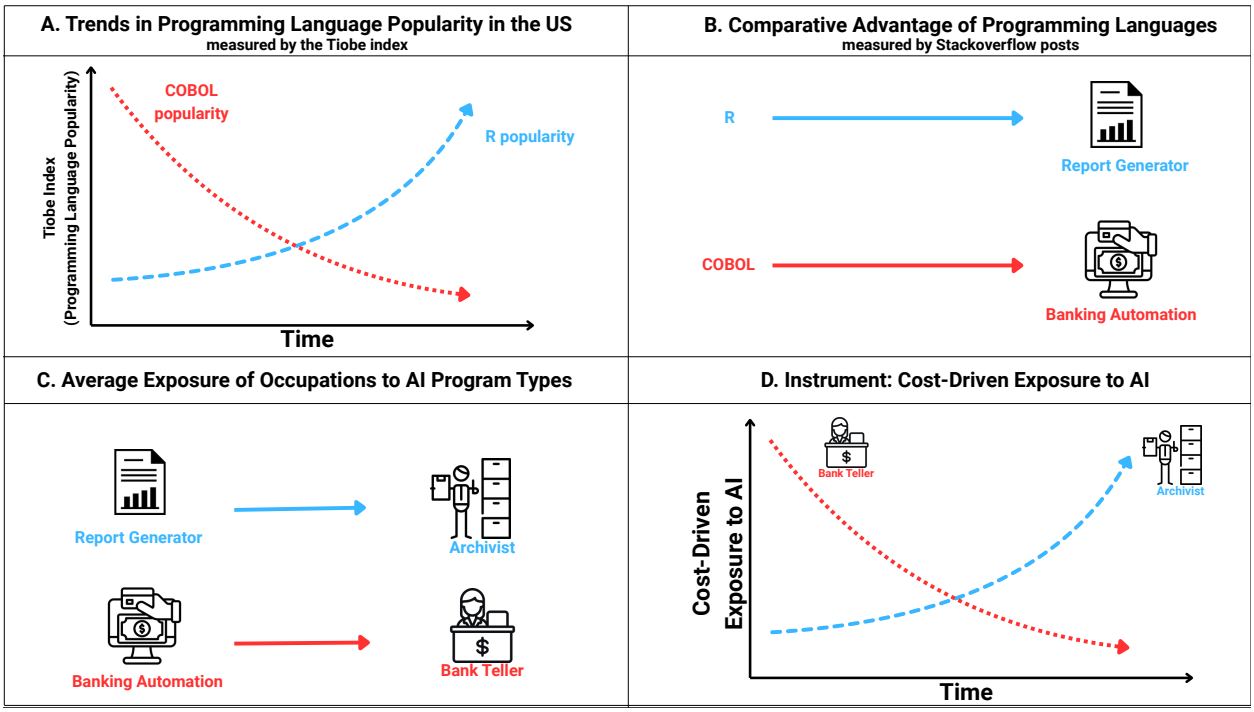
Exogenous variation: changes in the cost of creating software. I construct an instrument based on differential exposure to cost-saving U.S.-based libraries, tools, or resources. The key idea is that languages with large, active communities provide extensive libraries, tutorials, forums, and off-the-shelf code, allowing developers to reuse scripts rather than build them from scratch. As U.S. communities around a language grow, this stock of reusable knowledge expands, allowing Brazilian firms to develop AI software at lower cost.[33]

The cost advantage created by a growing programming-language community does not affect all occupations equally. Since programming languages are tailored to specific applications, these savings benefit some applications more than others (Nanz and Furia 2015, Rabah et al. 2010, Meyerovich and Rabkin 2013). For instance, as the U.S. Matlab community grows, it becomes easier to build AI engineering tools; if HTML grows, the cost advantage instead favors web applications. As some languages gain popularity in the U.S., AI tools in Brazil become disproportionately cheaper to build in certain application areas.

[33]The availability of resources and libraries has been shown to decrease the time and cost of software development (Delorey et al. 2007, Krein et al. 2010, Tambad et al. 2020, Vijayaraghavan et al. 2022).

As a consequence, occupations whose tasks overlap with those applications become relatively more exposed to AI software.

Figure 10: **Illustration of the Instrument**



Notes: This figure illustrates how the instrument is constructed.

Example: COBOL for financial transactions and R for statistical analysis. To illustrate how the instrument works, consider two programming languages: COBOL and R. COBOL (Common Business-Oriented Language) was created in the 1950s for business applications and remains widely used in financial systems and government operations (Ovide 2022, Bailey 2024, Gershgorn 2020, Sullivan 2025). Over 80% of in-person transactions and 95% of ATM swipes still rely on COBOL-based systems (Hartman 2015). Despite this prevalence, COBOL's popularity has declined since its inception. According to the TIOBE Index, which measures the popularity of programming languages, COBOL dropped from a top-12 programming language in 2003 to a top-22 in 2023 (TIOBE 2025). This decline led to a shortage of experienced programmers, outdated documentation, and difficulties integrating with modern languages. Overall, these constraints make software development for the financial sector relatively more costly (Navot 2024).

Following a different trend from COBOL, R, created in the 1990s for statistical analysis, has experienced a remarkable increase in popularity. According to the TIOBE Index, R is the 15th most popular programming language in 2025 (TIOBE 2025). As a result, developing AI software for statistical analysis is comparatively less costly due to the large pool of available resources for R.

Therefore, because writing software in COBOL is relatively more expensive than writing it in R, occupations linked to banking automation, such as bank tellers or bank managers, are less exposed to AI than those focused on statistical reports, like archivists. The instrument identifies the effect of AI on employment by comparing employment growth between these two types of occupations, as illustrated in Figure 10. The identifying assumption is that the decline in COBOL and the rise of R in the U.S. are unrelated to the labor market for archivists and bank tellers in Brazil.

Instrument definition. Using these insights, the ease of building AI applications targeting occupation o in year t is given by

$$\textit{AI Ease-of-Dev}_{o,t} = \sum_{l,p} \frac{\textit{Resources in US}_{l,t-2} \times \textit{Specialization}_{l,p} \times \overline{\textit{AI Exposure}}_{p,o,t-2}}{N_l N_p}, \quad (4)$$

where l indexes programming languages and p denotes a software technical class from the NIIP classification. The instrument combines three components: (i) the time-varying availability of resources for language l in the U.S. lagged 2 years, $\textit{Resources in US}_{l,t-2}$; (ii) the degree to which language l is suited to technical class p, $\textit{Specialization}_{l,p}$; and (iii) the baseline relevance of technical class p to tasks in occupation o, $\overline{\textit{AI Exposure}}_{p,o,t-2}$, calculated using data lagged 2 years. I normalize by the number of languages N_l and technical classes N_p.

Number of webpages in the U.S. mentioning a programming language is a proxy for the availability of resources. I proxy resource availability, $\textit{Resources in US}_{l,t-2}$, with the number of U.S.-based Google search hits for programming language l, lagged by two years.[34] A search hit count reflects the number of English-language webpages that Google.com associates with the term. This measure reflects the volume of U.S.-based web content related to the language, such as libraries, tutorials, books, and forum discussions. A greater volume of resources lowers the marginal cost of creating AI software by increasing

[34]Appendix A.4 summarizes the TIOBE dataset, which is the source for the number of hits of each programming language.

the productivity of programmers, who can reuse available scripts rather than starting from scratch (Delorey et al. 2007, Krein et al. 2010, Tambad et al. 2020, Vijayaraghavan et al. 2022). Supporting this link, Table A10 in the Appendix shows that higher lagged U.S. search hits predict more Brazilian AI software registrations using that language. I lag the search hits by two years to reflect the information available at the start of a software project and to mitigate reverse causality.[35]

Stack Overflow reflects the specialization of programming languages. The second component of the instrument, $Specialization_{l,p}$, measures the specialization of programming language l for the NIIP's software technical class p. To measure this, I calculate the share of posts on Stack Overflow with tags of programming language l and technical class p.[36]

$$Specialization_{l,p} = \frac{Stack\ Overflow\ posts\ tagging\ language\ l\ and\ technical\ class\ p}{Stack\ Overflow\ posts\ tagging\ language\ l} \quad (5)$$

The component $Specialization_{l,p}$ reflects two aspects of software development: a language's built-in comparative advantage for certain applications and the availability of libraries. Some programming languages are designed for specific tasks: HTML and JavaScript for web development, Swift for app development, Fortran for scientific computing, SQL for data management, and R for data science, among many others. Beyond design intent, historical factors such as legacy code, rich libraries, and active user communities reinforce specialization. For example, C++ and C# dominate game development (Wijdan 2020), while Python and Perl are widely used in bioinformatics (Fourment and Gillings 2008).[37] Table A5 shows that, on average, the leading language in each application area accounts for over 42% of related software projects, demonstrating that programming languages are indeed specialized in practice.

[35] According to statistics discussed in Section B.4, 45% of software projects take more than 2 years to finish.

[36] Stack Overflow is a popular question-and-answer website for computer programmers. Users create posts there asking for support on their code. When creating a post, users have to choose tags that reflect the nature of their post. Figure A16 lists the top 25 most popular tags in Stack Overflow. Those tags reflect the programming language used, the type of code written, and the overall application of the software. To link tags in Stack Overflow to NIIP's technical classes, I created a cross-walk. Section A.5, in the Appendix, provides summary statistics of Stack Overflow posts.

[37] According to Meyerovich and Rabkin (2013), the availability of libraries, community support, and legacy code largely determines why certain fields converge on a few languages. Rabah et al. (2010) emphasizes the inherent trade-offs across programming languages, which prevent any single one from dominating all applications.

Average AI exposure by technical class. The final component of the instrument connects technical classes to occupations. Specifically, $\overline{AI\ Exposure}_{p,o,t-2}$ is the average text similarity between AI software in technical class p and the task descriptions of occupation o, calculated using only AI software registered before $t-2$. Limiting the calculation to lagged years ensures that these exposure weights are predetermined and not mechanically correlated with AI adoption. Moreover, it also captures the evolving applications of AI.[38]

This component captures two dimensions of the relationship between technical classes and occupations. First, some AI technology classes are more relevant to certain occupations than others. For instance, archivists are more exposed to technical classes related to report generation than to those involving banking automation. Second, it reflects the breadth of exposure: some occupations are more broadly exposed to AI across many technical classes and, therefore, should be more affected by a decrease in the implementation cost of AI. For example, archivists, with high similarity scores across multiple classes, are more exposed than secretaries, whose average exposure is lower.

Aggregating to the same level of the AI exposure measure. Since the AI exposure measure is built using all AI software developed up to time t, I aggregate the AI Ease-of-Development in the same way:

$$AI\ Ease\text{-}of\text{-}Dev\ IV_{o,t} = \sum_{u=2003}^{t} AI\ Ease\text{-}of\text{-}Dev_{o,u} \tag{6}$$

Therefore, $AI\ Ease\text{-}of\text{-}Dev\ IV_{o,t}$ captures the ease of developing AI software up to year t.

Identifying variation. The instrument exploits differential exposure across occupations to an exogenous decline in the cost of developing AI software. The identifying assumption is a parallel-trends condition: in the absence of these cost shocks, the labor market outcomes of high- and low-exposure occupations would have followed similar trajectories over time (Goldsmith-Pinkham et al. 2020). This assumption ensures that any divergence in outcomes can be attributed to the differential effect of AI development costs, rather than to pre-existing trends or omitted shocks.

[38] In the robustness section, I show that using only AI software created before 2003 to construct the instrument does not change the results.

Validation. The main identification concern is that common shocks might drive both U.S. programming-language popularity and Brazilian labor-market outcomes, creating spurious correlations. To address this, Table A9 in the Appendix reports a placebo test regressing Brazilian labor market outcomes on AI exposure instrumented with the Ease-of-Development instrument between 1995 and 2002, a period known as the "AI winter" when few AI applications reached the market (Toosi et al. 2021, Chauvet 2018, Sevilla et al. 2022, Metz 2021). Since AI adoption was negligible then, any correlation would reflect common shocks rather than AI's effects. The results show no systematic relationship during this period, supporting the view that the instrument affects Brazilian markets only through AI adoption. Section 6 provides additional validation by controlling for U.S. labor market outcomes, using an instrument orthogonal to non-AI software, running regressions at the sector-occupation level with sector-year fixed effects, and controlling for trends with lagged dependent variables. The main conclusions remain robust across all specifications.

5 Results

In this section, I show that AI increases employment among young, less-educated, inexperienced, and low-ability workers, which in turn lowers average wages through a change in worker composition. Yet these aggregate effects hide substantial heterogeneity. Employment gains are concentrated in production roles tied to machine operation, while administrative occupations experience declines in employment. I show that these patterns are consistent with AI replacing occupations that process or organize information while expanding and deskilling occupations that use information to take action and make decisions.

5.1 AI increases employment of low-skilled workers

AI ease-of-development instrument predicts AI exposure. Table 2 reports the first-stage results under various sets of controls. Column 2, which presents the baseline specification, shows that a one–standard-deviation increase in the AI ease-of-development instrument increases AI exposure by approximately 0.78 standard deviations. The F-statistic is far above the threshold of 10, indicating that the instrument comfortably passes the weak instruments test.

Tables A10 and A11 in the Appendix clarify where the instrument gets its identifying variation. Table A10 shows that when a programming language becomes more popular

Table 2: **First-Stage: Ease-of-Development Increases AI Exposure**

	(1)	(2)	(3)
	AI Exposure	*AI Exposure*	*AI Exposure*
AI Ease-of-Dev IV	0.862***	0.780***	0.784***
	(0.00978)	(0.0101)	(0.0103)
N	45,336	45,336	45,336
R^2	0.751	0.770	0.780
F	7015.7	6291.7	5127.0

Notes: This table presents the estimates of the first-stage. The instrument is given by Equation (6) and the endogenous variable is the exposure to AI, defined in Equation (1). All columns include occupation and year fixed effects. Column 2 adds a 1-digit occupation-year fixed effect, and Column 3 a 2-digit occupation-year fixed effect. Standard errors are in parentheses. Standard errors are clustered at the occupation level. $*p < 0.10, **p < 0.05, ***p < 0.01$.

in the United States, Brazilian developers write more software in that language, both in levels and as a share of total filings. Table A11 shows that a language's Stack Overflow specialization, described in Equation (5), predicts which languages dominate within each technical class in Brazil. Together, these patterns support the interpretation that U.S. shocks to language popularity reduce development costs for specific AI applications, changing both the composition of software built and the set of occupations that face greater AI exposure.

AI leads to net employment gains. Table 3 shows the effect of AI on employment under different sets of controls. Across all specifications, AI exposure raises employment. In column 2, the baseline specification, a one-standard-deviation increase in AI exposure raises employment by 1.9%. Table A12 in the Appendix decomposes this net employment gain: AI increases both hiring and separation rates, but hiring rises by more, generating positive net job growth.

The positive employment effect is robust across specifications. Relative to the baseline in column 2, column 1 removes the year-by-1-digit-occupation controls. Column 3 adds year-by-sector fixed effects, where sector is defined as the primary two-digit industry for each occupation (accounting for 63% of occupational employment on average), to capture sectoral shocks that might drive the results.[39] Column 4 includes year-by-2-digit-occupation fixed effects. Column 5 combines year-by-sector and year-by-2-digit-occupation fixed effects.

[39]I construct the year–sector fixed effect by first calculating, for each 6-digit occupation, employment shares across 2-digit National Classification of Economic Activities sectors. I then assign each occupation to the sector with the highest share and use this dominant sector to define the sector–year fixed effect. Because the occupational classification is narrow, the dominant sector accounts for 63 percent of employment on average.

In all these specifications, the estimated coefficients are not statistically different from one another.

Table 3: **AI Increases Employment**

	(1)	(2)	(3)	(4)	(5)
	log(Employment)	*log(Employment)*	*log(Employment)*	*log(Employment)*	*log(Employment)*
Panel A: 2SLS					
AI Exposure	0.0229***	0.0194***	0.0202***	0.0158**	0.0177***
	(0.00600)	(0.00722)	(0.00624)	(0.00622)	(0.00631)
Panel B: OLS					
AI Exposure	0.0204***	0.0147**	0.0166***	0.0127**	0.0127**
	(0.00549)	(0.00583)	(0.00586)	(0.00563)	(0.00593)
N	45,336	45,336	45,119	45,336	45,119
Fixed Effects Interacted with Year:					
1-Digit Occ.	-	✓	✓	-	-
Main Sector	-	-	✓	-	✓
2-Digits Occ.	-	-	-	✓	✓

Notes: This table reports estimates of the effect of AI on log employment, following the specification (2). Panel A presents the 2SLS estimates using the AI ease-of-development instrument defined in Equation (6). Panel B shows the corresponding OLS estimates. All columns include year and occupation fixed effects. All columns except Column 1 include lagged employment and exposure to U.S. AI patents as a control. Column 1 includes only year and occupation fixed effects. Column 2 presents the baseline specification, adding year-by-1-digit-occupation fixed effects. Column 3 adds year-by-sector fixed effects to the baseline, where sector is defined as the primary two-digit industry for each occupation. Column 4 adds year-by-2-digit-occupation fixed effects. Column 5 combines year-by-sector and year-by-2-digit-occupation fixed effects. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10, ^{**}p < 0.05, ^{***}p < 0.01$.

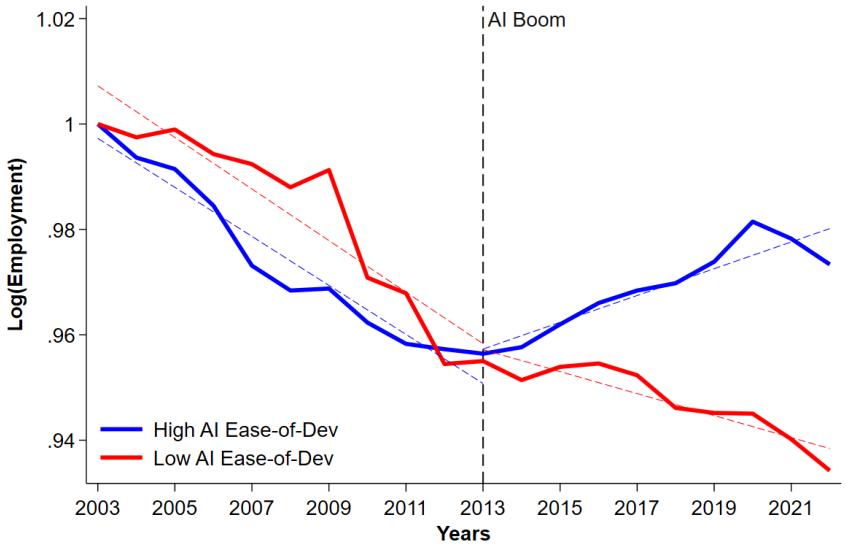
The OLS estimate in Table 3 is slightly smaller than the 2SLS estimate. One interpretation is that AI development in Brazil is largely supply-driven: domestic IT firms develop applications based on technological breakthroughs abroad rather than designing software in response to local labor market conditions. As a result, AI exposure mainly reflects foreign cost shocks, which are captured by the instrument, and is only weakly correlated with domestic labor demand shocks that would introduce bias, explaining the similarity between OLS and 2SLS results.

Figure 11 illustrates the identifying variation by plotting residual log employment for occupations in the top and bottom deciles of the long difference in the AI ease-of-development instrument, after removing the variation explained by the controls.[40] Before 2010, both groups followed parallel trends, supporting the identifying assumption. Beginning in 2013, coinciding with the AI boom and deep-learning breakthroughs, employment in high-exposure

[40]Figure A20 in the Appendix plots the correlation between employment growth and the AI Ease-of-Development instrument.

occupations (top decile) began to increase while employment in low-exposure occupations (bottom decile) continued to follow the same trend.

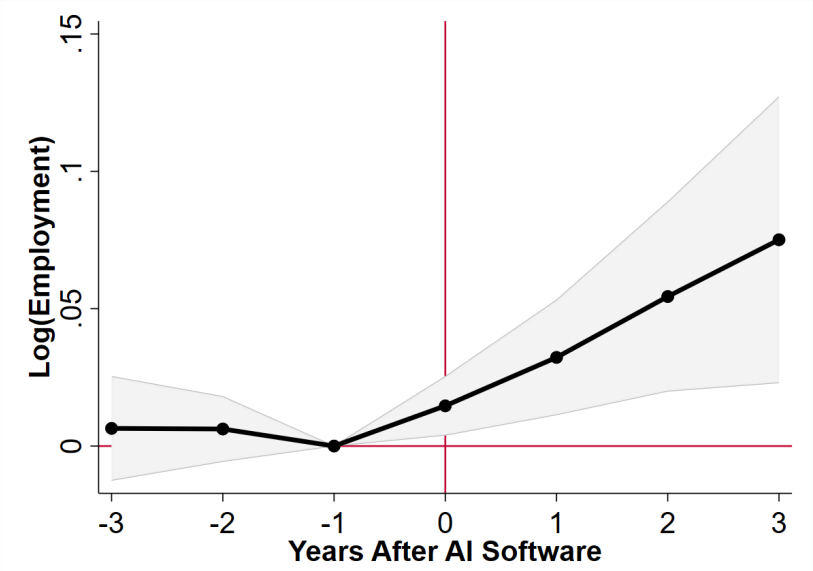
Figure 11: **Employment Grew Faster in Occupations with Higher AI Ease-of-Development after AI Breakthroughs**



Notes: This figure illustrates the identifying variation by plotting the component of log employment unexplained by the controls for occupations in the top and bottom deciles of growth in the AI ease-of-development instrument. To construct this figure, I compute the long difference in the instrument as $\Delta AI\ Ease\text{-}of\text{-}Dev\ IV_o = AI\ Ease\text{-}of\text{-}Dev\ IV_{o,2022} - AI\ Ease\text{-}of\text{-}Dev\ IV_{o,2003}$ and regress log employment on the full set of controls, including occupation fixed effects, one-digit-occupation-by-year fixed effects, and exposure to U.S. AI patents. The regression excludes both the instrument and the AI exposure. Using the residual, I construct a time-series for log employment by calculating for each occupation the deviations in employment from the initial value: $log(Employment_{o,0}) + \hat{\epsilon}_{o,t}$, where $\epsilon_{o,t}$ is the residual. The figure plots the log employment time-series from this regression for the top and bottom deciles of $\Delta AI\ Ease\text{-}of\text{-}Dev\ IV_o$.

No pre-trends support instrument validity. Figure 12 shows no relationship between current AI exposure and lagged employment, supporting the instrument's validity. AI has an immediate impact on the labor market following software registration, likely because firms register software shortly before adoption or commercialization. The effect grows over time: after three years, a one–standard–deviation increase in AI exposure raises employment by 7%. This delayed response reflects the time required for AI technologies to diffuse and reorganize production, consistent with historical patterns of technological adoption (Solow 1987, Brynjolfsson and Milgrom 2012, Brynjolfsson et al. 2021b, 2023).

Figure 12: **No Pre-Trends and Large Effects on the Long-Run**



Notes: This figure reports estimates of the dynamic effect of AI on employment, following the specification in Equation (3). All specifications use the AI ease-of-development instrument defined in Equation (6). Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, and exposure to U.S. AI patents. Standard errors are clustered at the occupation level. The shaded area reports 95% confidence intervals.

AI shifts employment toward younger, less educated, and inexperienced workers. Table 4 shows that AI reallocates the workforce toward younger, less educated, and less experienced workers. Specifically, column 1 shows a decrease in average age, column 2 a decline in average years of education, and column 3 a drop in average firm tenure. Table A14 in the Appendix shows that this compositional shift reflects larger employment gains among these demographic groups rather than declines in others.

AI reduces average wages due to the entry of lower-ability workers. Column 1 of Table 5 shows that a one-standard-deviation increase in AI exposure lowers average monthly wages by 2%. This decline comes from the entry of lower-wage workers into high-exposure occupations: as AI draws in younger, less educated, and less experienced workers, who generally earn less, it pulls the average wage down.[41]

To confirm that this wage decline reflects compositional changes rather than falling wages for existing workers, I decompose the effect into composition and price components. I estimate a Mincer regression relating individual wages to observable characteristics and worker fixed effects, then aggregate the worker fixed effects (which I call ability) and wage residuals

[41] Figure A21 shows the dynamic effects of AI on wages. The negative effects increase over time.

Table 4: **AI Increases Employment of Younger and Less Skilled Workers**

	(1) *log(Age)*	(2) *log(Yrs. Education)*	(3) *log(Tenure)*
Panel A: 2SLS			
AI Exposure	-0.00498^{***}	-0.00658^{***}	-0.0199^{***}
	(0.00112)	(0.00113)	(0.00591)
Panel B: OLS			
AI Exposure	-0.00415^{***}	-0.00521^{***}	-0.0208^{***}
	(0.000824)	(0.000783)	(0.00420)
N	45,336	45,332	45,336

Notes: This table reports estimates of the effect of AI on labor market outcomes, following the specification in equation (2). Panel A presents the 2SLS estimates using the AI ease-of-development instrument defined in Equation (6). Panel B shows the corresponding OLS estimates. Column 1 reports the effect of AI on the log of average age; column 2 on the log of average years of education; and column 3 on the log of average tenure at the current firm (in years). Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10$,$^{**}p < 0.05$,$^{***}p < 0.01$.

to the occupation level.[42] Because ability is fixed at the individual level, changes in average ability capture shifts in workforce composition, while changes in the average wage residual capture price movements unexplained by composition, i.e., wage per efficiency unit.

This decomposition confirms the composition story. Column 2 of Table 5 shows that AI lowers average ability, while column 3 shows that AI raises wages per efficiency unit, likely reflecting higher labor demand. The decline in average wages thus arises entirely from compositional change: AI draws in workers who earn less based on both observable and unobservable characteristics. This pattern is consistent with AI complementing low-ability workers, as experimental evidence shows larger productivity gains for lower-skill workers than for experts (Kanazawa et al. 2022, Brynjolfsson et al. 2025, Gruber et al. 2020, Choi et al. 2023, Dell'Acqua et al. 2023, Noy and Zhang 2023, Peng et al. 2023).

This increased demand for low-ability workers also compresses the wage distribution within occupations. Column 4 of Table 5 shows that a one–standard–deviation increase in AI exposure lowers the within-occupation standard deviation of wages by 3.4%. As AI raises the relative demand for workers at the bottom of the wage distribution, it narrows wage gaps within occupations.

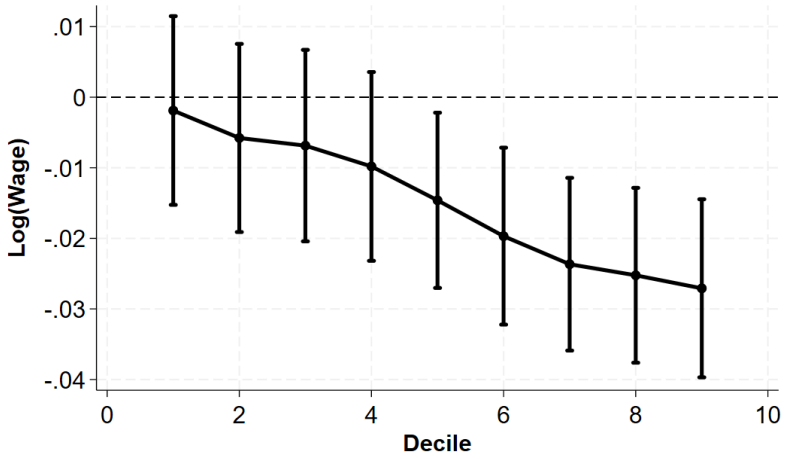
[42]The ability measure is calculated using the linked RAIS panel. Due to administrative lags in the release of this dataset, it only covers the period up to 2016, so these results capture only the beginning of the AI boom. Appendix B.5 provides a detailed description of the Mincer regression and estimation procedure.

Table 5: **AI Lowers Wages and Inequality From the Entry of Low-Ability Workers**

	(1) *log(Wage)*	(2) *Ability*	(3) *Wage Residual*	(4) *log(Std. Wage)*
Panel A: 2SLS				
AI Exposure	−0.0229***	−0.0196***	0.00493*	−0.0345***
	(0.00380)	(0.00401)	(0.00286)	(0.00992)
Panel B: OLS				
AI Exposure	−0.0191***	−0.0158***	0.00278	−0.0294***
	(0.00271)	(0.00376)	(0.00256)	(0.00661)
N	45,336	19,885	19,885	45,208

Notes: This table reports estimates of the effect of AI on labor market outcomes, following the specification in equation (2). Panel A presents the 2SLS estimates using the AI ease-of-development instrument defined in Equation (6). Panel B shows the corresponding OLS estimates. Column 1 reports the effect of AI on log average monthly earnings. Columns 2 and 3 show the effect of AI on the average ability and wage residuals, calculated using a Mincer regression at the occupation level and aggregated to occupations. Because ability and wage residuals require the panel structure of RAIS to be constructed, they are limited to the period from 2003 to 2016. Section B.5 explains in detail how these variables were calculated. In column 4 the variable of interest is the log standard deviation of wages within each occupation. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10, ^{**}p < 0.05, ^{***}p < 0.01$.

Figure 13: **AI Decreases Wages at the Top of the Wage Distribution**



Notes: This figure shows the effect of AI across different points in the wage distribution. To construct this figure, I compute wage-deciles within each occupation over the sample period. The figure plots the estimates of the effect of AI on the log of each wage decile. All specifications use the AI ease-of-development instrument defined in Equation (6). Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors are clustered at the occupation level. The bands report the 95% confidence intervals.

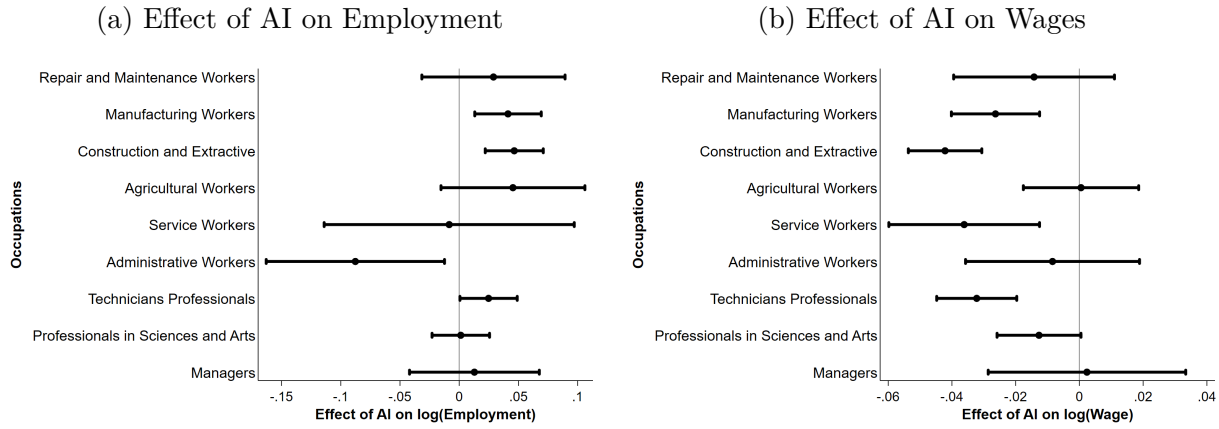
AI decreases inequality by lowering wages at the top of the distribution. To understand where in the wage distribution AI has its largest effects, Figure 13 plots the

estimated effect of AI on wages at each decile within occupations. A one–standard–deviation increase in AI exposure has no significant effect on wages in the lowest decile, while wages in the 9th decile fall by about 2.8%. AI thus decreases inequality by lowering wages at the top of the distribution, which is consistent with AI replacing expertise (Autor 2024, Autor and Thompson 2025).

5.2 AI in the Office and the Factory

AI replaces workers in the office but increases employment in the factory. Figure 14 shows how AI affects different types of occupations.[43] AI raises employment in production-related occupations, such as workers in agriculture, extraction, manufacturing, maintenance and technicians, by around 4%. In contrast, it reduces employment in administrative roles by 8%. Employment among managers and high-skilled professionals, such as engineers and chemists, remains largely unchanged.

Figure 14: **Differential Labor Market Effects of AI Across Occupations**



Notes: This figure reports estimates of the effect of AI on employment and wages for each one-digit occupational group. The model is: $y_{o,t} = \beta_{O(o)} AI\ Exposure_{o,t} + \mu_o + \mu_t + X'_{o,t}\theta + \epsilon_{o,t}$, where $O(o)$ denotes the broad occupational category of occupation o. All specifications use the AI ease-of-development instrument defined in Equation (6). In the first stage, the instrument is also interacted with a broad occupational group dummy. Standard errors are clustered at the occupation level. Bands represent 95% confidence intervals.

Figure 14b shows the effect of AI on wages across one-digit occupational categories. Managers are the only group experiencing wage increases, though the effect is not statistically significant. Production-related occupations—where AI increases employment—see the

[43] To create this figure, I allow the parameter of interest to vary by one-digit CBO occupational categories. The model is now given by $y_{o,t} = \beta_{O(o)} AI\ Exposure_{o,t} + \mu_o + \mu_t + X'_{o,t}\theta + \epsilon_{o,t}$, where $O(o)$ is the broad occupational class of occupation o.

largest wage declines. This pattern reflects the compositional shifts documented earlier: AI draws in lower-wage workers, pulling down average wages even as total employment rises.

To examine these patterns more systematically, I aggregate occupations into three broad categories. Production workers include agriculture, extraction, manufacturing, maintenance, and technicians—occupations where AI increases employment. Administrative workers form the second category, representing the office-based roles most vulnerable to automation. The third category, which I label professionals for simplicity, includes managers, skilled professionals, and service workers. Table 6 reports how AI's effects on employment and worker composition vary across these three groups.

Table 6: **AI Shrinks the Office and Expands the Factory**

	(1) *log(Employment)*	(2) *log(Age)*	(3) *log(Yrs. Education)*	(4) *log(Tenure)*	(5) *log(Wage)*
$\mathbb{I}\{Prod.\} \times AI\ Exposure$	0.0437^{***}	-0.00649^{***}	-0.00982^{***}	-0.0222^{***}	-0.0301^{***}
	(0.00997)	(0.00148)	(0.00198)	(0.00832)	(0.00457)
$\mathbb{I}\{Prof.\} \times AI\ Exposure$	0.0108	-0.00456^{***}	-0.00521^{***}	-0.0206^{***}	-0.0196^{***}
	(0.00861)	(0.00135)	(0.00109)	(0.00684)	(0.00466)
$\mathbb{I}\{Adm.\} \times AI\ Exposure$	-0.0878^{**}	0.00419	0.00239	0.0225	0.00789
	(0.0384)	(0.00566)	(0.00302)	(0.0295)	(0.0139)
N	45,336	45,336	45,332	45,336	45,336

Notes: This table reports 2SLS estimates of the effect of AI on labor market outcomes, allowing the parameter of interest to vary by production, professional, and administrative occupations. Each column uses the AI ease-of-development instrument defined in Equation (6). In the first stage, the instrument is also interacted with a dummy for each occupational group. Column 1 reports the effect of AI on log employment; column 2 on the log of average age; column 3 on the log of average years of education; column 4 on the log of average tenure at the current firm (in years); and column 5 on the log of average wage. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10,^{**} p < 0.05,^{***} p < 0.01$.

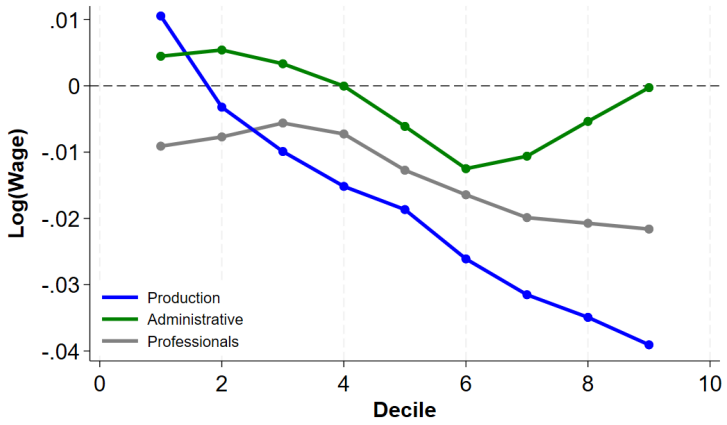
Table 6 shows that AI shrinks the office and expands the factory. In production occupations, AI increases employment among younger, less experienced, and lower-skilled workers. Because these new hires earn less than the existing workforce, this compositional shift lowers average wages and reduces within-occupation inequality. The magnitudes are substantial: a one-standard-deviation increase in AI exposure raises employment among production workers by 4.3% while lowering their average wage by 3%. By contrast, professionals experience an employment effect that is four times smaller.

In contrast, in administrative occupations, AI reduces employment without significantly changing workforce composition. A one-standard-deviation increase in AI exposure reduces

employment among administrative workers by 8%, with no statistically significant effect on average wages.

Employment gains in production are skewed toward low-wage workers. To understand how these compositional shifts translate into wage effects, Figure 15 examines AI's impact across the wage distribution separately for production, administrative, and professional workers.[44] This extends the earlier distributional analysis in Figure 13 by allowing AI's effects to differ across occupational categories.

Figure 15: **AI Compresses the Wage Distribution in Production and Hollows It Out in the Office**



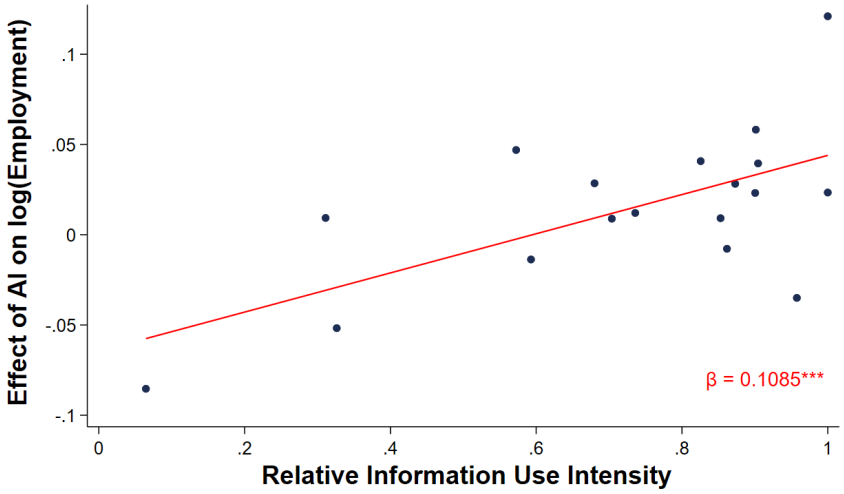
Notes: This figure shows the effect of AI across different points in the wage distribution for production, administrative, and professional workers. To construct this figure, I compute wage-deciles within each occupation over the sample period. The figure plots the estimates of the effect of AI on the log of each wage decile allowing the effect of AI to differ by production, administrative, and professionals. All specifications use the AI ease-of-development instrument defined in Equation (6). Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable.

Among production workers, AI compresses the wage distribution: a one-standard-deviation increase in AI exposure raises wages at the bottom decile by 1% while reducing wages at the ninth decile by 4%. This result suggests that, in production, AI acts as a skill substitute, lowering the expertise threshold required to perform tasks that previously demanded extensive training or experience (Autor 2024, Autor and Thompson 2025). At the bottom of the wage distribution, less-experienced workers become more productive as AI enables them to perform tasks they previously could not, raising their wages and expanding their employment. At the top of the wage distribution, experienced workers see their wages decline as AI diminishes the value of the expertise they have accumulated.

[44]To improve readability, confidence intervals are omitted from the figure.

AI hollows out the wage distribution among administrative workers. In administrative occupations, AI raises wages at the bottom of the distribution, lowers them in the middle, and has no discernible effect at the top—a pattern that mirrors the classic "hollowing out" or job polarization documented in the broader labor market (Autor et al. 2003, 2006, Autor and Dorn 2013, Goos et al. 2014, Michaels et al. 2014). However, rather than observing this polarization across occupations, as in the traditional literature where middle-skill jobs disappear while employment grows at the top and bottom, here the hollowing out occurs within the administrative occupation category itself.

Figure 16: **AI Automates Information Processing but Expands Information Use**



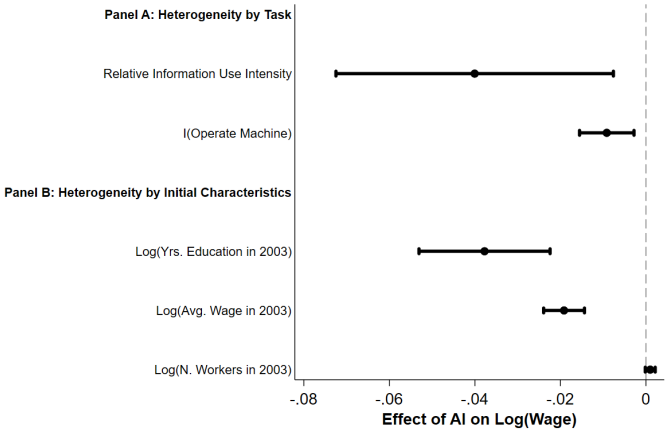
Notes: The figure plots the estimated effect of AI exposure on employment at the broad occupational level against an occupation's *relative information use intensity*. Each point corresponds to one of 20 broad occupational groups, narrower than the 1-digit level. The estimated effects are obtained from $y_{o,t} = \beta_{O(o)}\, AI\ Exposure_{o,t} + \mu_o + \mu_t + X'_{o,t}\theta + \epsilon_{o,t}$,where $O(o)$ denotes the broad occupational group containing occupation o. Relative information use intensity is constructed from task descriptions in the Brazilian CBO 2002 using a large language model to classify tasks into (i) information processing and recordkeeping, such as tasks related to filling out forms or maintaining inventory records, and (ii) information use to take action or make decisions, such as adjusting operating parameters based on readings or performing maintenance based on alerts. The relative information use intensity for broad occupation O is defined as $Relative\ Information\ Use\ Intensity_O =$ *shr. of tasks that use information/shr. of tasks that use or process information.*

Why does AI automate the office and expand the factory? According to Agrawal et al. (2019b), AI should increase employment in occupations whose tasks benefit from access to better operational information, such as supervisors or maintenance technicians, while replacing occupations whose tasks center on processing, organizing, or generating information, such as clerical workers or accounting clerks. To study this mechanism, Figure 16 plots the effect of AI on employment and wages at a broad occupational level against an occupation's relative information use intensity. I define this measure as the share of information-related

tasks that involve using operational information to guide decisions or actions, relative to all tasks involved in either using or processing operational information.[45] Higher values indicate occupations that are intensive users of operational information, such as managers, maintenance technicians, and machine operators, while lower values indicate occupations specialized in information processing and recordkeeping, including clerical workers, administrative assistants, and accounting clerks.

Figure 16 shows a strong positive relationship between the effect of AI on employment and an occupation's intensity in the use of operational information: employment grows in occupations that use information to guide actions but declines in occupations centered on processing or generating information. Table A13 in the Appendix shows that occupations that use information more intensively also experience a larger shift in worker composition toward low-skilled workers and a greater decline in wages. These patterns are consistent with AI disproportionately complementing less-experienced workers in information-use-intensive occupations, for whom limited experience previously constrained participation. One interpretation is that, by making high-quality information widely available, AI reduces the value of experience and provides expert knowledge to low-skilled workers.

Figure 17: **AI Decreases Inequality Across Occupations**



Notes: This figure reports heterogeneity in the effect of AI on monthly earnings based on different characteristics of each occupation. To construct it, I augment the baseline model by including an interaction term: $y_{o,t} = \beta AI\ Exposure_{o,t} + \beta^{hetero} AI\ Exposure_{o,t} \times x_o + \mu_o + \mu_t + X'_{o,t}\theta + \epsilon_{o,t}$, where x_o denotes an occupation-level characteristic. Panel A uses task-based indicators constructed from occupational descriptions: *Relative Information Use Intensity*, and *I(Operate Machine)* equals one if the occupation involves operating machinery in sectors like manufacturing, agriculture, or construction. Panel B calculates heterogeneity using characteristics of the occupation measured at the beginning of the period. All specifications use the AI ease-of-development instrument defined in Equation (6). Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Shaded areas represent 95% confidence intervals. Standard errors are clustered at the occupation level.

[45]This index is constructed using an LLM. I provide the details of its construction in Appendix B.6.

AI decreases inequality across occupations. Beyond reshaping wage distributions within occupations, AI also compresses the wage structure across occupations. Figure 17 shows that AI reduces wages more sharply in occupations that started with higher average education and wages in 2003, causing the occupational wage structure to compress. This convergence occurs through the same deskilling mechanism documented earlier: Figures A22a and A22b in the Appendix show that high-wage, high-education occupations experienced the largest influx of younger and less-educated workers in response to AI. As AI enables less-qualified workers to enter roles once dominated by high-skilled individuals, it erodes the wage premium these occupations once commanded. AI thus reduces inequality on two margins: it compresses wages within production occupations by raising pay at the bottom while lowering it at the top, and it compresses wages across occupations by disproportionately reducing pay in historically high-wage roles.

6 Robustness

In Section 5, I showed that AI increases employment among young, low-education, and inexperienced workers, which in turn lowers average wages due to a change in the composition of workers. In this section, I show that this result is robust to several tests: controlling for U.S. labor market outcomes; controlling for non-AI software; using an instrument orthogonal to non-AI software exposure; running regressions at the sector-occupation level; constructing the instrument using pre-period AI software; weighting regressions by initial employment; removing controls; and weighting AI exposure by software importance.

Control for Labor Market Outcomes in the U.S. One concern is that trends in U.S. programming languages may reflect global labor market shocks that also affect Brazil, creating spurious correlations. To address this, Panel A of Table 7 controls for the same labor market outcome used on the left-hand side, but measured in the U.S. using data from the Current Population Survey. This adjustment accounts for common global shocks to employment, education, experience, and wages. The results are unchanged: AI continues to increase employment among younger, less educated, and less experienced workers, and to lower average wages.

Control for non-AI software. To ensure that results isolate the effect of AI software from general software adoption, Panel B of Table 7 controls for exposure to non-AI software,

constructed similarly to the AI measure. The point estimates remain consistent with AI increasing employment and shifting the workforce toward lower-skilled workers; however, due to larger standard errors, the employment effect is not statistically significant.

AI exposure orthogonal to non-AI software. Another way to ensure that the results are not driven by exposure to software in general is to construct an instrument based on variation in AI exposure that is orthogonal to exposure to non-AI software. To do this, I first compute the average similarity between each occupation o and the technical classes p. I then regress $\overline{AI\ Exposure}_{p,o}$ on $\overline{Non\text{-}AI\ Exposure}_{p,o}$ and use the residual from this regression to construct the instrument. This residual isolates the component of AI exposure unrelated to general software exposure. Panel C of Table 7 presents the results using this alternative instrument. The conclusions remain qualitatively similar: AI increases employment and shifts the composition of the workforce toward lower-skilled workers. However, the effect on wages is no longer statistically significant due to larger standard errors, but it is still negative.

Regression at the occupation-industry, as in Webb (2020). In Panel D, I re-estimate the baseline specification at the occupation–industry level. Occupations are defined using the 6-digit CBO classification, and industries follow the 4-digit CNAE2 (*Classificação Nacional de Atividades Econômicas*) classification. Given the narrowness of the Brazilian occupational codes, this approach introduces substantial sparsity, with many occupation–industry cells missing from the data. To account for sector-specific shocks, I include industry-year fixed effects and occupation–industry fixed effects in addition to the baseline controls. Despite these adjustments and the resulting loss in precision, Panel D shows that the main conclusions remain robust.

Table 7: **Robustness: AI Increases Employment of Low-Skilled Workers**

	(1) *log(Employment)*	(2) *log(Age)*	(3) *log(Yrs. Education)*	(4) *log(Tenure)*	(5) *log(Wage)*	(6) *log(Std. Wage)*
Panel A: Control for U.S. Labor Market Outcome						
AI Exposure	0.0285^{***}	-0.00667^{***}	-0.00362^{***}	-0.00497^{***}	-0.0180^{***}	-0.00383
	(0.00889)	(0.00131)	(0.00178)	(0.00448)	(0.00442)	(0.0113)
Panel B: Control for Non-AI Software						
AI Exposure	0.0150	0.00718^{**}	-0.0114^{***}	-0.00304	-0.0229^{**}	-0.0442
	(0.0245)	(0.00339)	(0.00305)	(0.0175)	(0.0111)	(0.0312)
Panel C: Residual AI Exposure						
AI Exposure	0.0409^{***}	-0.00478^{**}	-0.00729^{***}	-0.0155	-0.0103	-0.0204
	(0.0117)	(0.00196)	(0.00174)	(0.0103)	(0.00634)	(0.0147)
Panel D: Sector-Occupation Level						
AI Exposure	0.0350^{***}	-0.00519^{***}	-0.00484^{***}	-0.00341^{*}	-0.0115^{***}	0.0156^{***}
	(0.00178)	(0.000351)	(0.000386)	(0.00180)	(0.000939)	(0.00544)
Panel E: Instrument with Pre-2003 AI Exposure						
AI Exposure	0.0134	-0.00357^{***}	-0.00602^{**}	-0.0225^{***}	-0.0138^{***}	-0.0279^{**}
	(0.00867)	(0.00135)	(0.00124)	(0.00692)	(0.00449)	(0.0109)
Panel F: Control for Trends						
AI Exposure	0.0200^{***}	-0.00251^{***}	-0.00314^{***}	-0.0107^{***}	-0.0101^{***}	-0.00958
	(0.00617)	(0.000773)	(0.000718)	(0.00369)	(0.00247)	(0.00690)
Panel G: Employment Weighted						
AI Exposure	0.0294^{***}	0.00264^{***}	-0.00223^{**}	-0.0164^{***}	-0.0207^{***}	-0.0403^{***}
	(0.00947)	(0.000774)	(0.00112)	(0.00543)	(0.00411)	(0.0110)
Panel H: No Controls						
AI Exposure	0.0428^{***}	-0.00558^{***}	-0.0116^{***}	-0.0205^{***}	-0.0184^{***}	-0.0119^{*}
	(0.00519)	(0.000792)	(0.00108)	(0.00379)	(0.00222)	(0.00614)
Panel I: Importance-Weighted AI Exposure						
AI Exposure	0.0226^{***}	-0.00410^{***}	-0.00386^{***}	-0.0144^{**}	-0.0166^{***}	-0.0157
	(0.00769)	(0.00127)	(0.00132)	(0.00651)	(0.00458)	(0.0107)

Notes: This table reports robustness checks of the effects of AI software exposure on labor market outcomes, following the specification in equation (2). Panel A controls for the U.S. version of each outcome variable to account for global labor market shocks. Panel B adds as control exposure to non-AI software, constructed using text similarity between the description of non-AI software and the description of each occupation's task. Panel C instruments AI exposure using variation orthogonal to non-AI software exposure. Panel D runs the baseline regression at the occupation–industry level, including industry-year and occupation–industry fixed effects. Panel E instruments AI exposure using only AI software developed before 2003 to construct the instrument. Panel F includes three lags of the dependent variable to control for pre-trends. Panel G reports employment-weighted regressions using 2003 employment as weights. Panel H excludes both the lagged dependent variable and the U.S. AI exposure control. Panel I shows the results weighting each software registration by the number of workers at the firm that developed it. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable, unless otherwise noted. Standard errors, clustered at the occupation level, are reported in parentheses. $^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$.

Instrument with Pre-2003 AI Software. In the baseline instrument described in equation (6), I use lagged AI software to compute each occupation's exposure to different AI technical classes. This approach allows me to capture the evolving range of AI applications that emerged during the period of analysis. In Panel E of Table 7, I instead construct the instrument using only AI software created before 2003. The results remain qualitatively unchanged. The effect of AI on employment is not statistically significant due to larger standard errors, but the coefficient is still positive.

Control for trends in labor market outcomes. To account for potential differential trends across occupations, Panel F of Table 7 augments the baseline controls by including three years of lagged values of the dependent variable. The results remain robust: I still find that AI increases employment, decreases wages, and shifts the composition of the workforce toward younger, less-educated, and less-experienced workers.

Employment-weighted regressions. Panel G of Table 7 reports results from baseline regressions weighted by 2003 employment. I find that AI increases employment among young, low-education, and inexperienced workers, which in turn lowers average wages due to a change in the composition of workers.

Regression without controls. Panel H of Table 7 presents results from regressions that exclude both the lagged dependent variable and controls for exposure to U.S. AI patents. The conclusions are still the same.

Importance-weighted AI exposure. The baseline AI exposure measure gives the same weight to all AI software. In practice, some tools are used more than others and matter more for the labor market. To capture this, I build an AI exposure measure that weights each technology by the size of the firm that developed it:

$$AI\ Importance\text{-}Weighted\ Exposure_{o,t} = \sum_{s} \rho_{o,s} log(N.\ Workers)_{t(s),s} \mathbb{I}_s \left\{ t(s) \leq t \right\}, \quad (7)$$

where $log(N.\ Workers)_{t(s),s}$ is the log of the number of workers at the firm that developed software s. Because of limits in the administrative data, the measure includes only software developed through 2016. Panel I of Table 7 shows that the main results stay the same.

7 Conclusion

This paper studies the effect of artificial intelligence on the labor market using a novel dataset of software registrations from Brazil's National Institute of Industrial Property (NIIP). The dataset contains detailed information on AI software developed in Brazil, including its intended applications and technical characteristics. I construct an occupation-level measure of AI exposure by computing text similarity between software descriptions and occupational tasks. To address endogeneity in AI adoption, I use an instrumental variable strategy that leverages variation in software development costs driven by trends in programming language popularity.

I find that, on average, AI increases employment but lowers wages, driven by rising employment among younger, less-educated, and less-experienced workers. Within occupations, AI reduces wage inequality by eroding gains from experience. Across occupations, however, the effects diverge: AI displaces administrative workers while expanding employment in production roles. These findings help reconcile competing views in the literature—confirming displacement effects in some settings while also presenting evidence that AI can complement labor and increase employment.

These results imply different policy responses across countries. Because AI displaces some workers while expanding employment for others, its net effect depends on a country's occupational mix. In economies with a large share of production workers, AI is likely to raise employment as workers move into production roles. In contrast, in economies dominated by administrative jobs, AI may reduce employment and force displaced workers to seek jobs in smaller production sectors or among professionals. Governments should therefore ease barriers to worker mobility across occupations, sectors, and regions. Where production sectors are large, this reallocation should mainly move workers from administrative jobs into production. Where agriculture and manufacturing are small, policy should instead support transitions from administrative work into professional roles.

This paper leaves several open questions for future research. Where do the low-skilled workers who gain employment through AI adoption come from? Given their age, AI may draw high-school–age workers into the labor market. Earlier entry could affect human capital accumulation and long-run growth. Another open question concerns administrative workers displaced by AI. What becomes of these workers, and how likely are they to transition into jobs that use AI? Answering these questions is essential for policy design and for understanding the long-run effects of AI.

References

Acemoglu, D., D. Autor, J. Hazell, and P. Restrepo (2022): "Artificial Intelligence and Jobs: Evidence from Online Vacancies," *Journal of Labor Economics*, 40, 293–340.

Acemoglu, D. and P. Restrepo (2018): "Artificial Intelligence, Automation, and Work," in *The Economics of Artificial Intelligence: An Agenda*, National Bureau of Economic Research, Inc, NBER Chapters, 197–236.

——— (2019): "The Wrong Kind of AI? Artificial Intelligence and the Future of Labor Demand," Working Paper w25682, National Bureau of Economic Research, Cambridge, MA, received May 30, 2019; accepted Nov 18, 2019.

——— (2020): "Robots and Jobs: Evidence from US Labor Markets," *Journal of Political Economy*, 128.

Adams, J., M. Fang, Z. Liu, and Y. Wang (2024): "The Rise of AI Pricing: Trends, Driving Forces, and Implications for Firm Performance," Working Paper Series 2024-33, Federal Reserve Bank of San Francisco.

Aghion, P., S. Bunel, X. Jaravel, T. Mikaelsen, A. Roulet, and J. Søgaard (2025): "How Different Uses of AI Shape Labor Demand: Evidence from France," *AEA Papers and Proceedings*, 115, 62–67.

Agoro, H. (2025): "Reducing Downtime in Production Lines Through Proactive Maintenance Strategies," .

Agrawal, A., J. Gans, and A. Goldfarb (2018): *Prediction Machines: The Simple Economics of Artificial Intelligence*, Harvard Business Review Press.

——— (2019a): "Prediction, Judgment, and Complexity: A Theory of Decision-Making and Artificial Intelligence," in *The Economics of Artificial Intelligence: An Agenda*, University of Chicago Press.

Agrawal, A., J. S. Gans, and A. Goldfarb (2019b): "Artificial Intelligence: The Ambiguous Labor Market Impact of Automating Prediction," NBER Working Papers 25619, National Bureau of Economic Research, Inc.

——— (2019c): "Exploring the impact of artificial Intelligence: Prediction versus judgment," *Information Economics and Policy*, 47, 1–6, the Economics of Artificial Intelligence and Machine Learning.

Alam, M. F., A. Lentsch, N. Yu, S. Barmack, S. Kim, D. Acemoglu, J. Hart, S. Johnson, and F. Ahmed (2024): "From Automation to Augmentation: Redefining Engineering Design and Manufacturing in the Age of NextGen-AI," Working paper, Massachusetts Institute of Technology, Cambridge, MA.

Alaveras, G. and B. Martens (2015): "International Trade in Online Services," JRC Working Papers on Digital Economy 2015-08, Joint Research Centre.

Aoun, M. R. E., L. N. Tidjon, B. Rombaut, F. Khomh, and A. E. Hassan (2022): "An Empirical Study of Library Usage and Dependency in Deep Learning Frameworks," *arXiv preprint*.

Arora, A. and A. Gambardella (2005): "The Globalization of the Software Industry: Perspectives and Opportunities for Developed and Developing Countries," in *Innovation Policy and the Economy, Volume 5*, National Bureau of Economic Research, Inc, NBER Chapters, 1–32.

Associação Brasileira das Empresas de Software (ABES) (2024): "Por que registrar meu software?" Accessed: 2024-11-14.

Aum, S. and Y. Shin (2024): "Is Software Eating the World?" Working Paper 32591, National Bureau of Economic Research.

——— (2025): "The Labor Market Impact of Digital Technologies," *Federal Reserve Bank of St. Louis Review*, 107, 85–108, accessed: 2025-05-29.

Autor, D. (2024): "Applying AI to Rebuild Middle Class Jobs," Working Paper 32140, National Bureau of Economic Research.

Autor, D., C. Chin, A. Salomons, and B. Seegmiller (2024): "New Frontiers: The Origins and Content of New Work, 1940–2018*," *The Quarterly Journal of Economics*, 139, 1399–1465.

Autor, D. and A. Salomons (2018): "Is Automation Labor Share–Displacing? Productivity Growth, Employment, and the Labor Share," *Brookings Papers on Economic Activity*, 1–63.

Autor, D. and N. Thompson (2025): "Expertise," NBER Working Papers 33941, National Bureau of Economic Research, Inc.

Autor, D. H. and D. Dorn (2013): "The Growth of Low-Skill Service Jobs and the Polarization of the US Labor Market," *American Economic Review*, 103, 1553–1597.

Autor, D. H., L. F. Katz, and M. S. Kearney (2006): "The Polarization of the U.S. Labor Market," *American Economic Review*, 96, 189–194, papers and Proceedings.

Autor, D. H., F. Levy, and R. J. Murnane (2003): "The Skill Content of Recent Technological Change: An Empirical Exploration," *The Quarterly Journal of Economics*, 118, 1279–1333.

Babina, T., A. Fedyk, A. He, and J. Hodson (2024): "Artificial intelligence, firm growth, and product innovation," *Journal of Financial Economics*, 151, 103745.

Bailey, C. (2024): "COBOL is a Cockroach: Why We Won't Get Rid of It Anytime Soon," `https://www.faircom.com/learn/blog/cobol-is-a-cockroach-why-we-wont-get-rid-of-it-anytime-soon`, accessed: 16 Feb 2025.

Barbosa, F., A. Souza, A. Souza, P. Nascimento, and A. Baganha (2022): "Análise Exploratória dos Registros de Software do Instituto Nacional da Propriedade Industrial (INPI) de 2018 a 2020," *Conjecturas*, 22, 765–777.

Barth, E., J. C. Davis, R. B. Freeman, and K. McElheran (2023): "Twisting the demand curve: Digitalization and the older workforce," *Journal of Econometrics*, 233, 443–467.

Benhanifia, A., Z. B. Cheikh, P. M. Oliveira, A. Valente, and J. Lima (2025): "Systematic review of predictive maintenance practices in the manufacturing sector," *Intelligent Systems with Applications*, 26, 200501.

Benz, S. and A. Jaax (2022): "The costs of regulatory barriers to trade in services: New estimates of ad valorem tariff equivalents," *Economics Letters*, 212, 110057.

Borboni, A., K. V. V. Reddy, I. Elamvazuthi, M. S. Al-Quraishi, E. Natarajan, and S. S. A. Ali (2023): "The Expanding Role of Artificial Intelligence in Collaborative Robots for Industrial Applications: A Systematic Review of Recent Works," *Machines*, 11, 111.

Boustan, L. P., J. Choi, and D. Clingingsmith (2022): "Automation After the Assembly Line: Computerized Machine Tools, Employment and Productivity in the United States," Working Paper 30400, National Bureau of Economic Research.

Bresnahan, T. (2021): *Artificial Intelligence Technologies and Aggregate Growth Prospects*, Cambridge University Press, 132–170.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei (2020): "Language Models are Few-Shot Learners," .

Brynjolfsson, E., W. Jin, and K. McElheran (2021a): "The power of prediction: predictive analytics, workplace complements, and business performance," *Business Economics*, 56, 217–239.

Brynjolfsson, E., W. Jin, and X. Wang (2023): "Information Technology, Firm Size, and Industrial Concentration," Working Paper 31065, National Bureau of Economic Research.

Brynjolfsson, E., D. Li, and L. Raymond (2025): "Generative AI at Work*," *The Quarterly Journal of Economics*, 140, 889–942.

Brynjolfsson, E. and K. McElheran (2016): "The Rapid Adoption of Data-Driven Decision-Making," *American Economic Review*, 106, 133–39.

——— (2019): "Data in Action: Data-Driven Decision Making and Predictive Analytics in U.S. Manufacturing," Working Paper 3422397, Rotman School of Management, University of Toronto, date written: July 6, 2019.

Brynjolfsson, E. and P. Milgrom (2012): "Complementarity in Organizations," in *The Handbook of Organizational Economics*, ed. by R. Gibbons and J. Roberts, Princeton University Press, chap. 1, 11–55.

Brynjolfsson, E., D. Rock, and C. Syverson (2021b): "The Productivity J-Curve: How Intangibles Complement General Purpose Technologies," *American Economic Journal: Macroeconomics*, 13, 333–72.

Capgemini (2020): "Artificial Intelligence (AI)-Driven Smart Factory Solution Provides Operators and Engineers with a New Level of Insight and the Ability to Adjust Production at a Moment's Notice," Tech. rep.

Chalfin, A., O. Danieli, A. Hillis, Z. Jelveh, M. Luca, J. Ludwig, and S. Mullainathan (2016): "Productivity and Selection of Human Capital with Machine Learning," *American Economic Review*, 106, 124–27.

Chauvet, J.-M. (2018): "The 30-Year Cycle In The AI Debate," .

Chen, R.-s., Y.-s. Tsai, and C.-c. Chang (2006): "Design and Implementation of an Intelligent Manufacturing Execution System for Semiconductor Manufacturing Industry," in *2006 IEEE International Symposium on Industrial Electronics*, vol. 4, 2948–2953.

Choi, J. H., D. Schwarcz, and K. E. Yeh (2023): "AI Assistance in Legal Analysis: An Empirical Study," Legal Studies Research Paper 23-22, University of Minnesota Law School.

Consoli (2025): "Inteligência artificial e termovisão reforçam eficiência da Copel e reduzem quedas de energia no Paraná," `https://aerp.org.br/redeaerp/inteligencia-artificial-e-termovisao-reforcam-eficiencia-da-copel-e-reduzem-quedas-de-energia-no-parana/`, publicado por Redação, com texto de Flávia Consoli — Associação das Emissoras de Radiodifusão do Paraná.

Danieli, O. (2026): "Skill-Replacing Technology and Bottom-Half Inequality," Working paper, Tel Aviv University; CEPR; IZA, previously titled "Revisiting U.S. Wage Inequality at the Bottom 50%".

de Souza, G., J. S. Herbstman, and J. Mannion (2024): "How Demand for New Skills Affects Wage Inequality: The Case of Software Programmers," Working Paper Series WP 2024-19, Federal Reserve Bank of Chicago.

de Souza, G. and H. Li (2023): "Robots, Tools, and Jobs: Evidence from Brazilian Labor Markets," Working Paper Series WP 2023-42, Federal Reserve Bank of Chicago.

Dell'Acqua, F., E. McFowland III, E. Mollick, H. Lifshitz-Assaf, K. C. Kellogg, S. Rajendran, L. Krayer, F. Candelon, and K. R. Lakhani (2023): "Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality," Working Paper 24-013, Harvard Business School, Boston, MA.

Delorey, D., C. Knutson, and S. Chun (2007): "Do Programming Languages Affect Productivity? A Case Study Using Data from Open Source Projects," 8–8.

Elsafty, A., M. Shaaban, A. El-Naggar, M. Okasha, and M. Elnaggar (2024): "AI-Driven Quality Management System for Sustainable Construction Projects: A Decision Support Model," *Energy Efficiency*.

Evans, R. and J. Gao (2016): "DeepMind AI Reduces Google Data Centre Cooling Bill by 40%," Accessed: 2025-05-08.

Ezell, S. J., R. D. Atkinson, and M. A. Wein (2013): "Localization Barriers to Trade: Threat to the Global Innovation Economy," Technical Report –, Information Technology & Innovation Foundation, Washington, DC, available at `https://www2.itif.org/2013-localization-barriers-to-trade.pdf`.

Felten, E., M. Raj, and R. Seamans (2021): "Occupational, industry, and geographic exposure to artificial intelligence: A novel dataset and its potential uses," *Strategic Management Journal*, 42, 2195–2217.

Felten, E. W., M. Raj, and R. Seamans (2018): "A Method to Link Advances in Artificial Intelligence to Occupational Abilities," *AEA Papers and Proceedings*, 108, 54–57.

Fonseca, M. and S. B. Cohen (2024): "Can Large Language Models Follow Concept Annotation Guidelines? A Case Study on Scientific and Financial Domains," .

Forootan, M. M., I. Larki, R. Zahedi, and A. Ahmadi (2022): "Machine Learning and Deep Learning in Energy Systems: A Review," *Sustainability*, 14, 4832.

Fourment, M. and M. R. Gillings (2008): "A comparison of common programming languages used in bioinformatics," *BMC Bioinformatics*, 9, 82.

Frey, C. B. and M. A. Osborne (2017): "The future of employment: How susceptible are jobs to computerisation?" *Technological Forecasting and Social Change*, 114, 254–280.

Gao, X. and K. Das (2024): "Customizing Language Model Responses with Contrastive In-Context Learning," .

Gershgorn, D. (2020): "Our Government Runs on a 60-Year-Old Coding Language — And Now It's Falling Apart," `https://onezero.medium.com/our-government-runs-on-a-60-year-old-coding-language-and-now-its-falling-apart-61ec0bc8e121`, accessed: 16 Feb 2025.

Ghelani, H. et al. (2024): "AI-Driven Quality Control in PCB Manufacturing: Enhancing Production Efficiency and Precision," *ResearchGate Preprint*, accessed: 2025-05-07.

GitHub Staff (2024): "Octoverse: AI leads Python to top language as the number of global developers surges," GitHub Blog.

Gmyrek, P., J. Berg, and D. Bescond (2023): "Generative AI and jobs a global analysis of potential effects on job quantity and quality," Tech. rep.

Goldsmith-Pinkham, P., I. Sorkin, and H. Swift (2020): "Bartik Instruments: What, When, Why, and How," *American Economic Review*, 110, 2586–2624.

Goos, M., A. Manning, and A. Salomons (2014): "Explaining Job Polarization: Routine-Biased Technological Change and Offshoring," *American Economic Review*, 104, 2509–2526.

Graetz, G. and G. Michaels (2018): "Robots at Work," *The Review of Economics and Statistics*, 100, 753–768.

Gruber, J., B. R. Handel, S. H. Kina, and J. T. Kolstad (2020): "Managing Intelligence: Skilled Experts and AI in Markets for Complex Products," NBER Working Papers 27038, National Bureau of Economic Research, Inc.

Halterman, A. and K. A. Keith (2025): "Codebook LLMs: Evaluating LLMs as Measurement Tools for Political Science Concepts," *Political Analysis*, 1–17.

Hampole, M., D. Papanikolaou, L. D. Schmidt, and B. Seegmiller (2025): "Artificial Intelligence and the Labor Market," NBER Working Papers 33509, National Bureau of Economic Research, Inc.

Hansen et al. (2024): "Just Read the Codebook: Using Large Language Models for Sociopolitical Text Classification with Expert Annotations," .

Harms, P. and D. Shuvalova (2020): "Cultural distance and international trade in services: A disaggregate view," *Economic Systems*, 44, 100786.

Hartman, T. (2015): "COBOL Blues," `https://www.reuters.com/graphics/USA-BANKS-COBOL/010040KH18J/`, accessed: 16 Feb 2025.

Hémous, D., M. Olsen, C. Zanella, and A. Dechezleprêtre (2025): "Induced Automation Innovation: Evidence from Firm-Level Patent Data," *Journal of Political Economy*, 133, 1975–2028.

Hosseini, S. M. and G. Lichtinger (2025): "Generative AI as Seniority-Biased Technological Change: Evidence from U.S. Résumé and Job Posting Data," SSRN Working Paper No. 5425555.

Hubmer, J. and P. Restrepo (2021): "Not a Typical Firm: The Joint Dynamics of Firms, Labor Shares, and Capital–Labor Substitution," Working Paper 28579, National Bureau of Economic Research.

Hui, X., O. Reshef, and L. Zhou (2024): "The Short-Term Effects of Generative Artificial Intelligence on Employment: Evidence from an Online Labor Market," *Organization Science*, earlier version: CESifo Working Paper No. 10601.

Humlum, A. (2019): "Robot Adoption and Labor Market Dynamics," Job Market Paper, Princeton University.

Humlum, A. and E. Vestergaard (2025a): "Large Language Models, Small Labor Market Effects," Working Paper 33777, National Bureau of Economic Research.

——— (2025b): "The unequal adoption of ChatGPT exacerbates existing inequalities among workers," *Proceedings of the National Academy of Sciences*, 122, 2414972121–.

Ide, E. and E. Talamàs (2025): "Artificial Intelligence in the Knowledge Economy," *Journal of Political Economy*.

IFPE (2025): "EDITAL REI/IFPE nº 036, de 04 de agosto de 2025: Concurso Público para Provimento de Vagas para o Cargo de Professor/a de Ensino Básico, Técnico e Tecnólogo (EBTT) do IFPE," PDF, edital (Reitoria/IFPE) para concurso público de docentes EBTT.

Instituto Nacional da Propriedade Industrial (INPI) (2015a): *Campo de aplicação*, Instituto Nacional da Propriedade Industrial, atualizado em 09/04/2015.

——— (2015b): *Tipos de programas de computador*, Instituto Nacional da Propriedade Industrial, atualizado em 09/04/2015.

Jaravel, X., P. Aghion, C. Antonin, and S. Bunel (2020): "What Are the Labor and Product Market Effects of Automation? New Evidence from France," CEPR Discussion Papers 14443, C.E.P.R. Discussion Papers.

Junqueira Botelho, A. J., G. Stefanuto, and F. Veloso (2005): "The Brazilian Software Industry," in *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland, and Israel*, Oxford University Press.

Kalyani, A., T. Hassan, and P. Restrepo (2025): "New Technologies and Inequality," Working paper, Federal Reserve Bank of St. Louis, available at author's website.

Kanazawa, K., D. Kawaguchi, H. Shigeoka, and Y. Watanabe (2022): "AI, Skill, and Productivity: The Case of Taxi Drivers," CIRJE F-Series CIRJE-F-1202, CIRJE, Faculty of Economics, University of Tokyo.

Kaplan, J., S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei (2020): "Scaling Laws for Neural Language Models," .

Keleko, A. T., B. Kamsu-Foguem, R. H. Ngouna, and A. Tongne (2022): "Artificial intelligence and real-time predictive maintenance in industry 4.0: a bibliometric analysis," *AI and Ethics*, 2, 553–577.

Kogan, L., D. Papanikolaou, L. D. Schmidt, and B. Seegmiller (2023): "Technology and Labor Displacement: Evidence from Linking Patents with Worker-Level Data," Working Paper 31846, National Bureau of Economic Research.

Kong, A., S. Zhao, H. Chen, Q. Li, Y. Qin, R. Sun, X. Zhou, E. Wang, and X. Dong (2024): "Better Zero-Shot Reasoning with Role-Play Prompting," .

Krein, J., A. Maclean, C. Knutson, D. Delorey, and D. Eggett (2010): "Impact of Programming Language Fragmentation on Developer Productivity: A Sourceforge Empirical Study," *IJOSSP*, 2, 41–61.

Kumar, S., C. Y. Park, and Y. Tsvetkov (2023): "Gen-Z: Generative Zero-Shot Text Classification with Contextualized Label Descriptions," .

Li, Y., X. Dang, H. Tian, T. Sun, Z. Wang, L. Ma, J. Klein, and T. F. Bissyandé (2024): "An Empirical Study of AI Techniques in Mobile Applications," *arXiv preprint*.

Liu, J., Y. Qian, Y. Yang, and Z. Yang (2022): "Can Artificial Intelligence Improve the Energy Efficiency of Manufacturing Companies? Evidence from China," *International Journal of Environmental Research and Public Health*, 19, 2091.

Long, Y., X. Wu, Y. Zhang, X. Wen, Y. Zhou, and S. Hong (2025): "Copy-Paste to Mitigate Large Language Model Hallucinations," .

Mais Floresta (2023): "Pixlog lança tecnologia que revoluciona medição de madeira," `https://www.maisfloresta.com.br/pixlog-lanca-tecnologia-que-revoluciona-medicao-de-madeira/`, accessed: 2026-01-23.

Mao, Y., J. He, and C. Chen (2025): "From Prompts to Templates: A Systematic Prompt Template Analysis for Real-world LLMapps," .

Marr, B. and M. Ward (2019): *Artificial Intelligence in Practice: How 50 Successful Companies Used AI and Machine Learning to Solve Problems*, Hoboken, NJ: John Wiley & Sons.

Martinez, J. (2021): "Putty-Clay Automation," *CEPR Discussion*.

Maslej, N., L. Fattorini, R. Perrault, Y. Gil, V. Parli, N. Kariuki, E. Capstick, A. Reuel, E. Brynjolfsson, J. Etchemendy, K. Ligett, T. Lyons,

J. Manyika, J. C. Niebles, Y. Shoham, R. Wald, T. Walsh, A. Hamrah, L. Santarlasci, J. B. Lotufo, A. Rome, A. Shi, and S. Oak (2025): "The AI Index 2025 Annual Report," Annual report, AI Index Steering Committee, Institute for Human-Centered AI, Stanford University, retrieved 9 April 2025 from `https://hai-production.s3.amazonaws.com/files/hai_ai_index_report_2025.pdf`.

McElheran, K., J. F. Li, E. Brynjolfsson, Z. Kroff, E. Dinlersoz, L. S. Foster, and N. Zolas (2023): "AI Adoption in America: Who, What, and Where," Working Paper 31788, National Bureau of Economic Research.

McElheran, K., M.-J. Yang, Z. Kroff, and E. Brynjolfsson (2025): "The Rise of Industrial AI in America: Microfoundations of the Productivity J-curve(s)," Working Papers 25-27, Center for Economic Studies, U.S. Census Bureau.

Metz, C. (2021): *Genius Maker: The Engineers Who Created the Intelligence Revolution*, New York: Viking.

Meyerovich, L. A. and A. S. Rabkin (2013): "Empirical analysis of programming language adoption," in *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications*, New York, NY, USA: Association for Computing Machinery, OOPSLA '13, 1–18.

Michaels, G., A. Natraj, and J. van Reenen (2014): "Has ICT Polarized Skill Demand? Evidence from Eleven Countries over Twenty-Five Years," *Review of Economics and Statistics*, 96, 60–77.

Min, S., X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer (2022): "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" .

Miroudot, S., J. Sauvage, and B. Shepherd (2010): "Measuring the Cost of International Trade in Services," MPRA Paper 27655, University Library of Munich, Germany.

Murtaza, A., A. Saher, M. Zafar, S. Moosavi, M. Aftab, and F. Sanfilippo (2024): "Paradigm shift for predictive maintenance and condition monitoring from Industry 4.0 to Industry 5.0: A systematic review, challenges and case study," *Results in Engineering*, 102935.

Nanz, S. and C. A. Furia (2015): "A Comparative Study of Programming Languages in Rosetta Code," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, IEEE, 778–788.

Navot, G. (2024): "The 8 Hidden Costs of Legacy Mainframe Codes in the Finance and Banking Sector," `https://overcast.blog/the-8-hidden-costs-of-legacy-mainframe-codes-in-the-finance-and-banking-sector-51e4e1ee62e3`, accessed: 16 Feb 2025.

Noy, S. and W. Zhang (2023): "Experimental evidence on the productivity effects of generative artificial intelligence," *Science*, 381, 187–192.

Ovide, S. (2022): "The Long Life of COBOL: Why Banks and Governments Still Rely on It," `https://www.nytimes.com/2022/07/06/technology/cobol-jobs.html`, accessed: 16 Feb 2025.

Patel, H. and H. Ghelani (2024): "The Role of Artificial Intelligence (AI) in Quality Control of Industrial Products," ResearchGate, accessed: 2025-05-07.

Peng, S., E. Kalliamvakou, P. Cihon, and M. Demirer (2023): "The Impact of AI on Developer Productivity: Evidence from GitHub Copilot," .

Pixlog (n.d.): "Pixlog Manager," `https://pixlog.com.br/`, accessed: 2026-01-23.

Prytkova, E., F. Petit, D. Li, S. Chaturvedi, and T. Ciarli (2024): "The Employment Impact of Emerging Digital Technologies," CESifo Working Paper Series 10955, CESifo.

Rabah, S., J. Li, M. Liu, and Y. Lai (2010): "Comparative Studies of 10 Programming Languages within 10 Diverse Criteria – a Team 7 COMP6411-S10 Term Report," .

Reynolds, L. and K. McDonell (2021): "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm," .

Ruckdeschel, M. (2025): "Just Read the Codebook! Make Use of Quality Codebooks in Zero-Shot Classification of Multilabel Frame Datasets," in *Proceedings of the 31st International Conference on Computational Linguistics*, ed. by O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, Abu Dhabi, UAE: Association for Computational Linguistics, 6317–6337.

Schmitz, J. A. (2005): "What Determines Productivity? Lessons from the Dramatic Recovery of the U.S. and Canadian Iron Ore Industries Following Their Early 1980s Crisis," *Journal of Political Economy*, 113, 582–625.

Schwarz, C. (2019): "lsemantica: A command for text similarity based on latent semantic analysis," *The Stata Journal*, 19, 129–142.

Sevilla, J., L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos (2022): "Compute Trends Across Three Eras of Machine Learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 1–8.

Shi, W., X. Li, J. Zhang, and J. Sun (2023): "Design and Implementation of an AI-Based Quality Control System in Automotive Production," *Procedia Computer Science*, 228, 2322–2328.

Shinohara, E. and Y. Kawazoe (2026): "Efficient medical NER with limited data: Enhancing LLM performance through annotation guidelines," *International Journal of Medical Informatics*, 208, 106230.

Shojaeinasab, A. et al. (2022): "Intelligent Manufacturing Execution Systems: A Systematic Review," *Journal of Manufacturing Systems*, 62, 503–522.

Solow, R. (1987): "We'd Better Watch Out," *New York Times Book Review*.

Stuber, W. D. (1984): "Industrial Policy in the Field of Informatics in Brazil," *Michigan Journal of International Law*, 6, 303–.

Sullivan, M. (2025): "Elon Musk, Dogecoin, and the COBOL Programming Language," `https://www.fastcompany.com/91278597/elon-musk-doge-cobol-language`, accessed: 16 Feb 2025.

Tambad, S., R. Nandwani, and S. K. McIntosh (2020): "Analyzing programming languages by community characteristics on Github and StackOverflow," .

Tariq, U., A. Khan, S. Shah, M. Imran, et al. (2024): "A Novel Approach for Automated Quality Inspection Using Deep Learning-Based Computer Vision in Manufacturing Industry," *Electronics*, 13, 976.

Tbit (2024): "GroundEye – Rações," PDF brochure, produced by Tbit. PDF available on tbit.com.br (path includes /2024/08/).

TIOBE (2025): "TIOBE Programming Community Index," https://www.tiobe.com/tiobe-index/, accessed: 2025-07-18.

Toosi, A., A. G. Bottino, B. Saboury, E. Siegel, and A. Rahmim (2021): "A Brief History of AI: How to Prevent Another Winter (A Critical Review)," *PET Clinics*, 16, 449–469.

Torquato, P. (2022): "Importance of Registering a Software in Brazil," Accessed: 2024-11-14.

Ucar, A., M. Karakose, and N. Kirimça (2024): "Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends," *Applied Sciences*, 14, 898.

Vijayaraghavan, S., J. Song, T. Guan, S. Choi, and S. Kulkarni (2022): "Influence of Communication Among Shared Developers on the Productivity of Open Source Software Projects," .

Webb, M. (2020): "The Impact of Artificial Intelligence on the Labor Market," Retrieved from https://www.michaelwebb.co/webb_ai.pdf.

Wijdan, A. (2020): "Comparison of Programming Languages in Game Development," .

Xu, M., J. Liu, Y. Liu, F. X. Lin, Y. Liu, and X. Liu (2018): "A First Look at Deep Learning Apps on Smartphones," *arXiv preprint*.

Yang, F., Y. Liang, and X. Zhou (2023): "Learning Quality-Aware Representations for AI-Based Inspection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 10506–10514.

A Data

A.1 Developers use software registration to signal achievements

Figure A1: **LinkedIn posts by developers showcasing their NIIP software-registration certificates.**



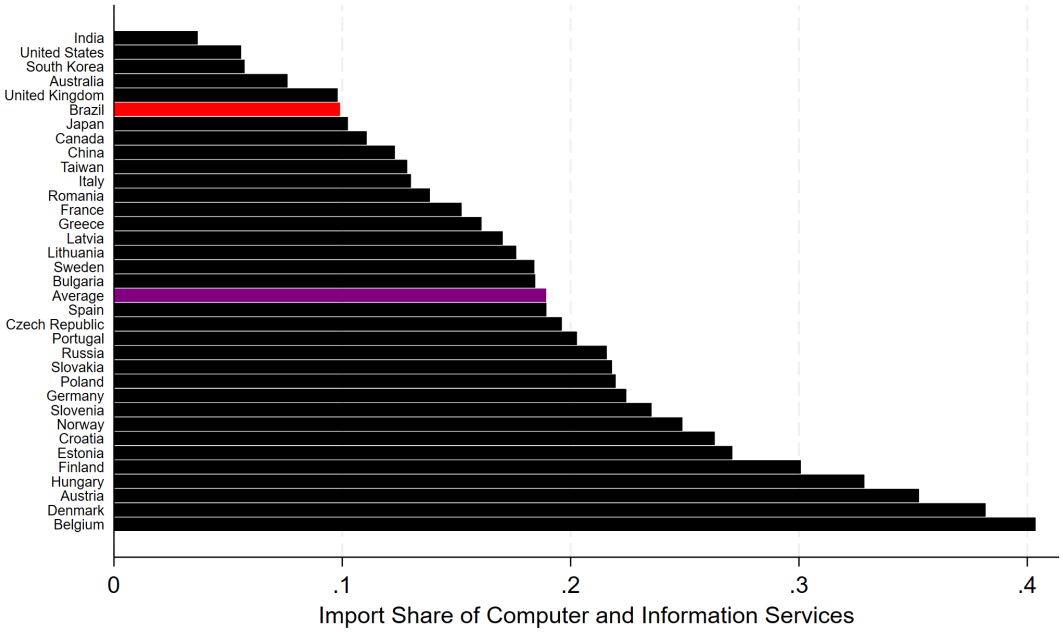
A.2 Most software used in Brazil is developed locally and therefore registered with the NIIP

Figure A2 reports the import share of computer services—a category that includes software publishing, programming, data processing, and all cross-border licensing fees for packaged software—using data from the WTO–OECD Balanced Trade in Services dataset combined

with value-added data from the PREDICT database.[46] For Brazil, the import share of computer services is only 9.3%, compared to an OECD average of roughly 19%. Because imported software is typically paid for through licensing fees and does not appear in Brazil's copyright registry, the low import ratio suggests that the vast majority of software used by Brazilian firms is developed locally. As a result, the NIIP database, which records nearly the entire universe of domestic software, captures most of the software actually used in Brazil.

Figure A2: **Import Share of Computer Services Across the Globe in 2015**

In Brazil, Most Software is Locally Developed



Notes: This figure shows the import share of computer services across countries in 2015. The import share is calculated by dividing the import in computer services, which includes license fees paid to other countries, by the value added in computer services. Data on imports of computer services comes from the WTO-OECD Balanced Trade in Services dataset. Data on value added of computer services comes from the PREDICT Dataset.

One potential caveat is that foreign software vendors might operate through Brazilian subsidiaries. In such cases, inter-company transfers are recorded as royalties rather than licensing fees, and the parent firm might bypass Brazil's requirement to register source code locally. To assess the relevance of this channel, Table A1 reports the share of multinational enterprises (MNEs) in Brazil's IT sector: fewer than 1 percent of firms are foreign-controlled. This negligible presence suggests that nearly all software used in the country is produced by domestically owned firms and thus captured by the NIIP registry.

[46]Due to data restrictions, I can't remove data processing and data center services from computer services. However, according to (Junqueira Botelho et al. 2005), the Brazilian software market accounts for 98% of the Brazilian market.

Table A1: **Share of Multinational Firms in IT Sectors**

	Share of Multinationals	*Employment-Weighted Share*
Information Technology Services	0.020%	0.025%
Information Service Provision Activities	0.038%	0.078%
Total	0.025%	0.035%

Notes: This table reports the share of foreign companies operating in the Brazilian IT sector in 2016. The sector *Information Technology Services* correspond to CNAE 2.0 code 62 and *Information Service Provision Activities* to CNAE code 63.

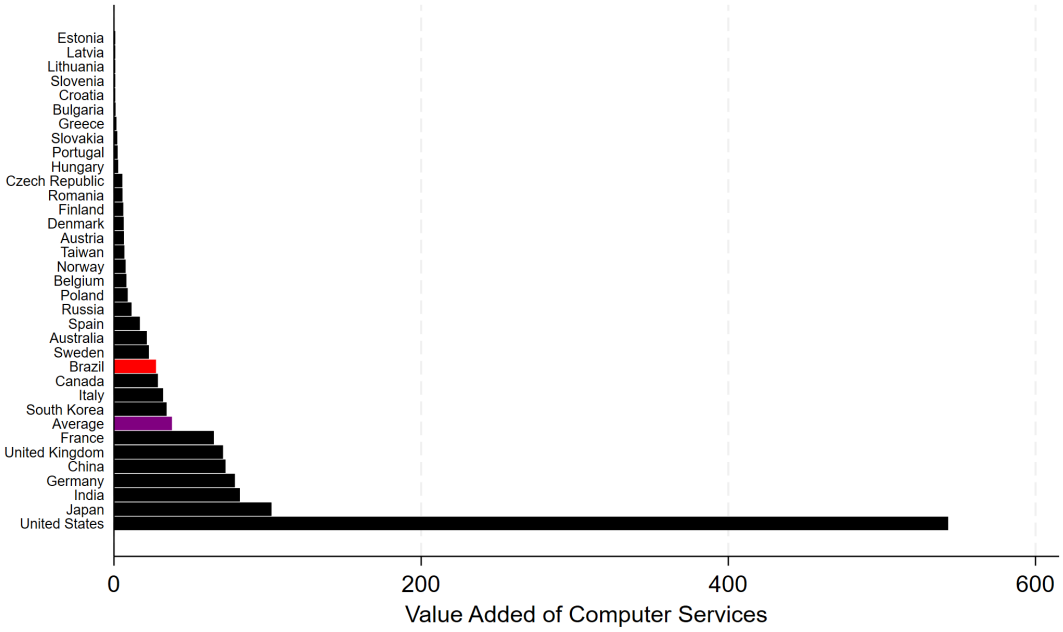
Three mutually reinforcing forces keep Brazil's import share of IT services exceptionally low. First, digital services tend to exhibit a strong home bias across countries. Alaveras and Martens (2015) shows that even in the United States, only the largest firms export online services, and that the preference for domestic providers is stronger for digital services than for physical goods. Trade costs in services are typically two to three times higher than in goods (Miroudot et al. 2010), partly due to cultural and regulatory frictions (Harms and Shuvalova 2020).

Second, Brazil reinforces this global home bias through explicit regulatory barriers. Local infrastructure mandates, data-localization requirements, and domestic-content rules in public procurement raise the effective trade cost of imported IT services to the equivalent of an 87 percent tariff—compared to an average of just 7 percent for goods trade (Ezell et al. 2013, Benz and Jaax 2022).

Third, Brazil has long fostered a large domestic software industry. Figure A3 shows that Brazil ranks as the 11th largest IT sector globally, ahead of developed countries such as Sweden, Australia, and Spain. As early as 2001, it ranked seventh worldwide, comparable to China and India (Junqueira Botelho et al. 2005). Large-scale public and private initiatives—such as social-security automation, electronic voting, and nationwide banking systems—generated sustained local demand (Arora and Gambardella 2005). As a result, 98 percent of software services in 2001 were supplied by Brazilian companies (Junqueira Botelho et al. 2005).

Taken together, the global difficulty of trading services, Brazil's unusually high regulatory barriers, and its mature domestic software industry explain why imported software plays only a minor role. Therefore, the NIIP registry provides near-census coverage of the code actually used within Brazilian firms.

Figure A3: **Value Added of Computer Services Across the Globe in 2015**



Notes: This figure shows value added of computer services across the globe in 2015 using data from the PREDICT Dataset in billions of 2015 dollars.

A.2.1 Survey with Industry Specialists

In this section I describe the results of a survey with industry specialists who have expert knowledge of the NIIP registration process. The survey delivers three main findings. First, software registration in the NIIP is widely recognized among both intellectual property lawyers and IT professionals, with the vast majority having direct experience with the registration process. Second, respondents confirm that most commercial software developed in Brazil is registered, with firms primarily motivated by intellectual property protection. Third, to the extent that selection exists, it is driven by small firms producing consumer-facing applications such as video games and mobile apps, categories that are unlikely to affect the labor market outcomes studied in this paper. These findings support the validity of using NIIP data to measure AI exposure in Brazilian workers.

Survey design. I conducted the survey with two groups that have detailed knowledge of the NIIP registration process: intellectual property lawyers and IT professionals. I identified potential respondents through an online search, constructing a database of 342 intellectual property law offices and 276 software development companies. I contacted these firms and of-

fices through email or LinkedIn between September and December of 2025. Each respondent was compensated for their participation.

In total, 68 people completed the survey: 41 intellectual property lawyers and 27 IT professionals. This gives a response rate of 11%. This small response rate comes from the fact that I contacted firms through publicly available email addresses, which often serve only customer inquiries rather than research requests. Individual lawyers, who typically manage their own correspondence, showed higher response rates than firms.

Survey respondents are experienced in their field. Table A2 presents summary statistics for the sample of respondents. Respondents average more than 10 years of experience in either IT or intellectual property law, with 86% holding at least a bachelor's degree. Among IT professionals, 81% work full-time in the IT sector, and 62% identify as programmers. Therefore, this is an experienced sample of respondents that are well-suited to assess the prevalence and patterns of software registration in Brazil.

Table A2: **Survey Summary Statistics**

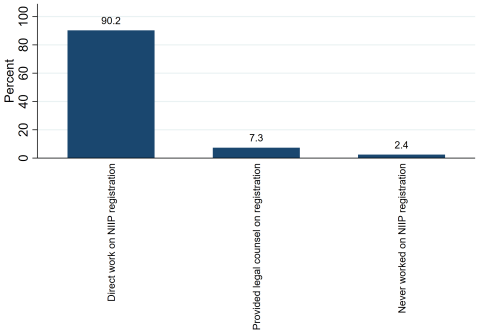
	Programmers	Lawyers	All
Average age	36.2	43.8	40.7
Average experience	12.9	18.9	16.6
Bachelor degree or more (%)	70.4	97.6	86.8

NIIP registration is recognized among specialists. The NIIP software registration process is widely known among industry specialists. Among IT professionals, 96% report being familiar with the registry. Among intellectual property lawyers, 90% have actively participated in at least one software registration, with the average lawyer having registered 37 distinct software programs. Among IT professionals, 77% report either having participated in a registration themselves or working at a firm that has registered software. This high level of familiarity indicates that the NIIP registry is a salient feature of Brazil's software industry, consistent with the institutional background described in Section 2.1.

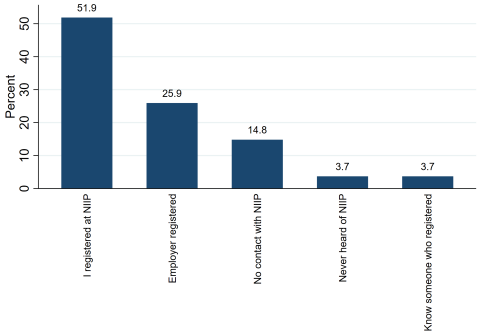
Figure A5a shows IT professionals' responses to the question: "How many commercial software programs do you believe have at least one version registered in the NIIP?" Seventy percent of respondents believe that firms register all or most of their commercial software,

Figure A4: **Experience of Respondents with a Software Registration**

(a) Intellectual Property Lawyer



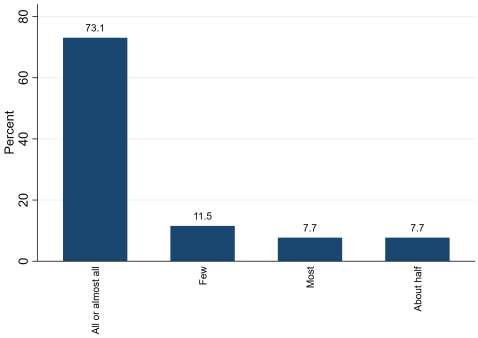
(b) IT Professional



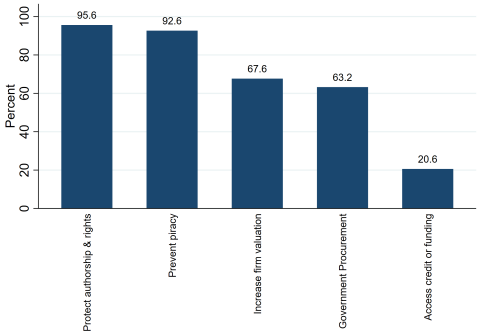
though 11% believe that only a few software programs are registered. This distribution suggests that while registration is the norm, some selection may exist.

Figure A5b displays the percentage of respondents who agree with various motivations for registering software in the NIIP. The dominant reasons are protecting intellectual property rights and preventing piracy, consistent with the main objectives of the NIIP.

(a) How many commercial software do you believe have at least one version registered in the NIIP?



(b) Why are firms registering software?



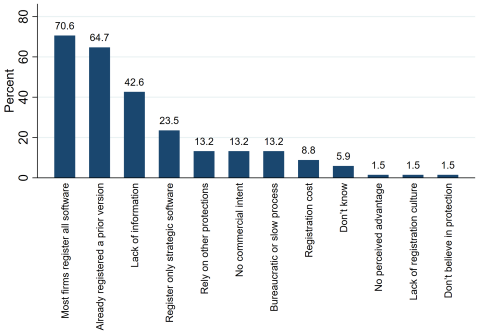
Selection is driven by small firms producing video games or cellphone apps. To understand which types of software might be underrepresented in the NIIP registry, I asked respondents about non-registration patterns. Figure A6a shows the reasons respondents cite

for why some software remains unregistered. The two most common explanations are that a previous version of the software has already been registered or that the firm is unaware of the NIIP. Notably, only 4% of IT professionals in the survey report being unfamiliar with the registry, suggesting that, although the lack of awareness explains why firms do not register, it is quantitatively a small factor.

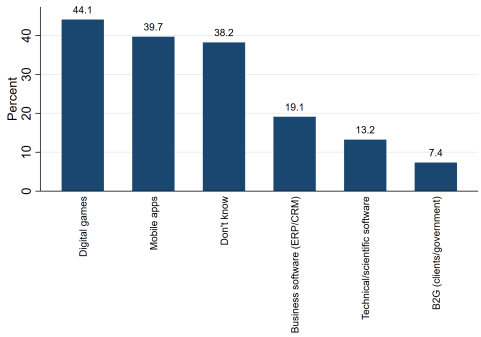
Figure A6b identifies which types of software are least likely to be registered. Respondents most frequently cite video games and mobile apps. Follow-up interviews with respondents reveal that developers of these consumer-facing applications perceive limited benefits from NIIP registration because distribution platforms (such as app stores) already provide intellectual property protections. Moreover, 83% of respondents agree that large companies are more likely to be aware of the NIIP and register their software than small firms.

Taken together, these findings indicate that any possible selection in the NIIP registry would be against small firms producing consumer-oriented applications. These categories represent a small share of enterprise software and are unlikely to significantly affect the labor market outcomes examined in this paper. The software that is used by firms, such as enterprise resource planning systems, manufacturing execution systems, and business intelligence tools, is comprehensively covered in the NIIP data.

(a) Why some firms don't register their software?



(b) What types of software do you think are less likely to be registered?



A.2.2 List of Broad Technical Classes and Application Domains

Table A3: Broad Technical Classes and Description

Code	Title	# Subcategories	Description
AP	Integrated Financial Suite	5	An Integrated Financial Suite is a comprehensive set of software tools designed to manage and streamline an organization's financial operations. It typically includes applications for planning budgets and projects, maintaining accounting records, performing audits, and overseeing internal controls. These tools work together to ensure financial accuracy, regulatory compliance, and strategic decision-making, providing a unified platform for managing the financial health and accountability of a business. Subcategories: Accounting, Application, Auditing, Control, and Planning.
AT	Automation Systems	8	Automation Systems are software-driven solutions that perform tasks with minimal human intervention, improving efficiency, consistency, and speed across various domains. This category spans office automation for routine administrative tasks, commercial and banking automation for streamlining business operations, and industrial automation for controlling manufacturing and production processes. It also includes specialized applications like automotive electronics, which manage vehicle systems. Subcategories: Automotive Electronics, Automation, Banking Automation, Commercial Automation, Industrial Automation, Manufacturing Automation, Office Automation, and Process Control.
AV	Resource Accounting	2	Resource Accounting tools monitor and evaluate how effectively software and hardware resources are utilized. This includes performance evaluation tools that test and measure system speed, responsiveness, and stability, as well as resource accounting tools that track the consumption of computing, energy, or other operational resources. These tools help identify inefficiencies, optimize performance, and support informed decision-making for system improvements and cost control. Subcategories: Performance Evaluation and Resource Accounting.
CD	Network Management	6	Network Management software enables the setup, monitoring, and coordination of computer networks and connected devices. It includes tools for managing local area networks (LANs), handling data communication between systems, and ensuring peripheral devices connect smoothly across the network. Terminal emulators allow users to access and control remote machines, while broader network management tools help maintain performance, security, and reliability. These systems are essential for keeping digital communication running smoothly, both within organizations and across distributed environments. Subcategories: Data Communication, Local Network, Network Management, Peripheral Management, Teleprocessing Monitor, and Terminal Emulators.
CT	Telecommunication Terminals	4	Telecommunication Terminals refer to systems and software that manage end-point communication devices and their supporting infrastructure. This includes telephone and telegraph switching systems that route calls and messages, as well as terminal operating systems that coordinate logistics and operations in terminal environments. Tools for function implementation and operations/maintenance management enhance performance and ensure system reliability. Together, these components optimize how terminals connect, operate, and are maintained within broader telecommunications networks. Subcategories: Function Implementation, Operations/Maintenance Management, Telephone/Telegraph Switching, and Terminal Operating System.
DS	Development Support Tools	8	Development Tools (Dev Tools) are essential software components that help programmers create, translate, and execute code efficiently. This category includes programming languages—both procedural and non-procedural—as well as the tools that transform source code into executable programs. Compilers, assemblers, and interpreters convert human-readable code into machine instructions, while pre-compilers, cross-compilers, and preprocessors adapt and optimize code for specific environments or hardware. Together, these tools form the technical foundation for software development, enabling developers to build, test, and maintain applications across platforms and architectures. Subcategories: Application Generator, Applications Development System, Computer Aided Software Engineering, Documentation Support, Programming Support, Routine Library, Systems Converter, and Systems Development Support Tool.
ET	Entertainment Software	4	Entertainment Software encompasses programs designed to engage users through interactive, visual, or immersive experiences. This includes video games, graphic art tools for creative expression, and virtual reality applications that simulate real or imagined environments. These tools are used for leisure, artistic development, and media production, blending technology with creativity to deliver engaging and often interactive content across platforms. Subcategories: Entertainment, Graphic Art, Video Game, and Virtual Reality.
FA	Office Applications	4	Office Applications are software tools designed to enhance productivity in everyday business and administrative tasks. This category includes word processors for document creation, spreadsheets for data analysis, and graphing tools for visualizing information. It also encompasses support tools like help desk software that assist in managing tasks and workflows. Together, these applications streamline communication, data handling, and problem-solving across a wide range of office environments. Subcategories: Electronic Spreadsheet, Graph Generator, Support Tool, and Word Processor.

Continued on next page

Code	Title	# Subcategories	Description
GI	Data Management	8	Data Management software supports the organization, storage, access, and integrity of digital information across its lifecycle. It includes tools for collecting and validating data, maintaining files, generating reports, and managing structured databases. Features like data dictionaries help define and standardize data use, while recovery tools ensure information can be restored after loss or corruption. The category also encompasses personal information managers and prototyping tools that aid in planning and interacting with data systems. Together, these tools enable individuals and organizations to handle data efficiently, securely, and reliably. Subcategories: Data Dictionary, Data Entry and Validation, Data Recovery, Database Manager, File Maintenance, Personal Information Manager, Report Generator, and Software Prototyping.
IA	AI Applications	3	AI Applications use intelligent algorithms to perform tasks that typically require human reasoning, learning, or perception. This includes expert systems that simulate decision-making in specialized fields, and natural language processing (NLP) tools that enable computers to understand, interpret, and generate human language in text or speech. Broadly, AI applications are used in areas such as healthcare, finance, customer service, and automation. Subcategories: Artificial Intelligence, Expert System, and Natural Language Processing.
IT	Measurement Tools	4	Measurement Tools includes software and systems designed to collect, analyze, and interpret data from various instruments. This includes general instrumentation software that controls and monitors devices, test and measurement tools used in scientific and industrial settings, and specialized biomedical instrumentation for healthcare applications. Analytical instrumentation further supports detailed examination and processing of data to derive meaningful insights. Together, these tools enable accurate and reliable measurement critical to research, diagnostics, and quality control across many fields. Subcategories: Analytical Instrumentation, Biomedical Instrumentation, Instrumentation, and Test and Measurement Instrumentation.
LG	Programming Language and Compilers	9	Programming Language and Compilers are essential software components that help programmers create, translate, and execute code efficiently. This category includes programming languages—both procedural and non-procedural—as well as the tools that transform source code into executable programs. Compilers, assemblers, and interpreters convert human-readable code into machine instructions, while pre-compilers, cross-compilers, and preprocessors adapt and optimize code for specific environments or hardware. Together, these tools form the technical foundation for software development, enabling developers to build, test, and maintain applications across platforms and architectures. Subcategories: Assembler, Compiler, Cross Compiler, Interpreter, Non-Procedural Language, Pre-Compiler, Pre-Processor, Procedural Language, and Programming Language.
PD	Data Protection	5	Data Protection software safeguards digital information from unauthorized access, loss, or corruption. This includes tools for securing data through encryption (cryptography), maintaining accuracy and consistency (data integrity), and controlling who can access or modify information (access control). Password managers help users create and store strong credentials securely, while broader security tools protect sensitive data across systems. Together, these solutions ensure the confidentiality, integrity, and availability of information in both personal and organizational settings. Subcategories: Access Control, Cryptography, Data Integrity, Password Manager, and Security and Data Protection.
SM	Simulation	4	Simulation software replicates real-world processes, systems, or environments to aid in analysis, training, or design. This category includes modeling tools for testing scenarios virtually, operating system simulators for educational or development purposes, and general simulators like those used in driving or flight training. It also encompasses engineering design and simulation tools such as CAE and CAM, which help optimize product designs and manufacturing processes. By allowing safe, cost-effective experimentation, simulation tools support better decision-making, improved performance, and innovation across various fields. Subcategories: Engineering Design/Simulation, Operating System Simulator, Simulation and Modeling, and Simulator.
SO	System Management	9	System Management software oversees the core functions that enable a computer or network to operate efficiently and reliably. It includes tools that manage hardware resources, coordinate input/output operations, and ensure seamless communication between system components. This category spans operating systems, device managers, user interfaces, and process controllers — all working together to allocate resources, run applications, and facilitate user interaction. System Management also covers lower-level interfaces for disks and networks, as well as control mechanisms that govern system behavior and automation. Subcategories: Basic Disk Interface, Command Processor, Communication Interface, Device Management, Input/Output Interface, Network Controller, Operating System, Process Controller, and User Interface Management.
TC	Image Processing	4	Image Processing software analyzes, enhances, and interprets visual data for a wide range of technical and scientific applications. It includes tools for digital image manipulation, pattern recognition, and operations research to extract meaningful information from images. These programs are used in fields such as medical imaging, remote sensing, and quality control, enabling automated analysis, improved accuracy, and deeper insights from visual inputs. Subcategories: Image Processing, Operations Research, Pattern Recognition, and Technical Scientific Application.

Continued on next page

Code	Title	# Subcategories	Description
TI	Telecom Systems	4	Telecom Systems support the exchange of information over distances through electronic means. This category includes technologies for data transmission, terminal devices that connect users to networks, and packet switching methods that efficiently route data across digital networks. Encompassing both hardware and software components, telecom systems form the foundation of modern communication—enabling voice, video, and data services across local and global networks. Subcategories: Data Transmission, Packet Switching, Telecommunication, and Terminals.
UT	Data Utilities	6	Data Utilities are supportive software tools designed to manage, process, and optimize data handling across computing systems. This includes utilities for compressing files, converting media formats, and transferring data between devices or networks. Tools like classifiers and interleavers help organize or structure data efficiently, while spool controllers manage queued tasks such as print jobs. Together, these utilities enhance system performance, streamline data workflows, and support seamless data exchange and transformation. Subcategories: Classifier/Interleaver, Data Compression, File Transfer, Media Conversion, Spool Controller, and Utilities.

Note: This table presents the code, title, number of subcategories, and description of each broad technical class. The description of each broad category was created by the authors using information from the original classification provided by the NIIP, Instituto Nacional da Propriedade Industrial (INPI) (2015a), and an inspection of the software in each classification.

Table A4: Broad Application Domains and Description

Code	Title	# Subcategories	Description
AD	Administration	11	The Administration application domain includes software that supports the management of public and private organizations. These systems handle strategic planning, operational control, and institutional development. Subfields include public administration, business management, production planning, personnel and material management, asset oversight, marketing, and office operations. Applications range from tools for organizing workflows and evaluating performance to systems for managing inventory, recruitment, sales strategies, and administrative communication. Subcategories: Organizational Development, Administrative Function and Planning, Modern Administration, Public Administration, Business Administration, Production Management, Personnel Management, Material Management, Asset Management, Marketing, Office Administration.
AG	Agriculture	14	The Agriculture application domain includes software that supports the planning, production, and management of agricultural systems. It encompasses crop cultivation, livestock farming, rural development, and agricultural policy. Applications address areas such as soil management, irrigation, pest control, and farm infrastructure through agricultural engineering tools. The domain also covers agricultural economics, aquaculture, and forestry, providing support for decision-making in resource use and production efficiency. Additionally, it includes systems for managing the extraction of plant and animal products, promoting sustainable use of natural resources. Subcategories: Agriculture and Rural Development, Agricultural Sciences, Agricultural Property Management, Agricultural Economics, Agricultural Systems, Agricultural Engineering, Soil Science, Plant Pathology, Crop Production, Animal Production and Veterinary Science, Forest Sciences, Aquaculture, Plant Extraction, Animal Extraction.
AH	Human Geography	6	The Human Geography application domain includes software that analyzes how human societies interact with space and place. It supports the study of social structures, cultural practices, urban and rural development, and the organization of communities. Applications cover topics such as migration, population distribution, urbanization, and public policy related to housing and infrastructure. This domain also incorporates tools for demographic analysis and urban geography, enabling users to model spatial patterns of human activity and assess their economic, social, and environmental impacts. Subcategories: Human Settlements, Urban Centers, Territorial Organization, Human Settlement Policies, Population Dynamics, Supporting Disciplines.
AN	Anthropology	7	The Anthropology application domain includes software that supports the study of human societies, cultures, and their evolution. It encompasses tools for analyzing social organization, community development, cultural traditions, and belief systems across different population groups. This domain integrates physical and cultural anthropology with sociology and the study of religion, enabling the examination of identity, social behavior, and institutional change. Applications often assist in ethnographic research, demographic studies, and comparative social analysis, offering insights into both historical and contemporary patterns of human life. Subcategories: Society and Social Structure, Social Development, Social Groups, Culture, Religion, Anthropology, Sociology.

Continued on next page

Code	Title	# Subcategories	Description
BL	Biology	8	The Biology application domain includes software that supports the study of living organisms and biological systems at all levels of organization. It spans molecular to systemic processes, aiding research in areas such as gene expression, cellular structure, microbial behavior, organ function, and metabolic regulation. Key subfields include genetics, cytology, microbiology, anatomy, physiology, biochemistry, and biophysics. These applications are used for modeling biological functions, analyzing experimental data, and simulating biochemical and physiological systems, contributing to advancements in medicine, environmental science, and biotechnology. Subcategories: Biology, Genetics, Cytology, Microbiology, Anatomy, Physiology, Biochemistry, Biophysics.
BT	Botany	4	The Botany application domain includes software that supports the study of plant life, focusing on plant structure, function, classification, and ecological distribution. Applications range from tools for analyzing plant morphology and physiology to systems for studying vegetation patterns across biomes. The domain also includes software for economic botany, which evaluates the properties and uses of crops, fibers, medicinal plants, and other economically important species. Taxonomic tools support plant identification and classification, while phytogeographic systems help map and understand regional vegetation. These applications are essential for biodiversity research, agriculture, and sustainable natural resource management. Subcategories: Botany, Phytogeography, Economic Botany, Plant Taxonomy.
CC	Construction	10	The Construction application domain includes software that supports the design, planning, execution, and management of building projects. It covers residential, commercial, industrial, and large-scale public works, incorporating traditional and modern construction techniques such as masonry, concrete, prefabrication, and mechanized processes. Applications assist with structural design, project coordination, cost estimation, bidding, and contract compliance, especially in public infrastructure projects. Specialized systems address structural analysis, finishing works, ventilation and lighting, hydraulic structures, and soil mechanics. These tools ensure efficient, safe, and sustainable construction practices across all phases of the building lifecycle. Subcategories: Construction, Construction Processes, Construction Management, Public Works, Structural Systems, Building Components, Construction Techniques, Building Hygiene, Hydraulic Engineering, Soil and Earthworks.
CO	Communication	6	The Communication application domain includes software that facilitates the creation, analysis, and dissemination of knowledge, language, and culture. It integrates tools from philosophy, science, and linguistics to explore how meaning is constructed and shared. Applications support mass media and public discourse through journalism, advertising, public relations, and editorial design. The domain also includes artistic expression—such as photography, cinema, music, and literature—as well as historical research and cultural preservation. These systems enable both practical communication tasks and broader engagement with cultural and intellectual life. Subcategories: Philosophy, Science and Scientific Methodology, Linguistics, Communication and Media, Arts, History.
DI	Legal	3	The Legal application domain includes software designed to support the creation, interpretation, and enforcement of laws across various levels of governance. It covers legislative drafting, legal analysis, and the application of constitutional principles that structure political institutions and protect individual rights. This domain also encompasses a wide range of legal disciplines, including civil, labor, tax, environmental, commercial, and international law. Applications assist with case management, contract generation, legal research, and compliance monitoring. These tools serve both public institutions and private entities, promoting legal order, protecting rights, and enabling the resolution of disputes across societal sectors. Subcategories: Legislation and Legal Interpretation, Constitutional Law, Specialized Legal Disciplines.
EC	Economics	14	The Economics application domain includes software that analyzes how individuals, firms, and governments allocate resources and respond to incentives. It covers both theoretical modeling and empirical analysis across microeconomics, macroeconomics, and national accounting. Applications support the study of production, consumption, and pricing, as well as broader economic dynamics like inflation, growth, and business cycles. The domain also includes tools for regional development, trade policy, monetary systems, income distribution, and firm behavior. These systems help inform decision-making and policy across local, national, and global economic contexts. Subcategories: Economic Theory and Methodology, Microeconomic Analysis, Macroeconomic Theory, Economic Activity, National Accounts, Monetary Economics, Market Structures, Products, Economic Engineering and Dynamics, Regional and Urban Economics, Property and Land Structure, International Economics, Economic Policy, Business and Corporation Finances.
ED	Education	6	The Education application domain includes software that supports the delivery, administration, and innovation of learning across formal and informal settings. It covers all levels of education, from early childhood to postgraduate studies, as well as adult education and vocational training. Applications assist with curriculum development, instructional methods, educational management, and policy implementation. The domain also includes tools for distance learning, multimedia instruction, and community-based programs. These systems help tailor educational experiences to diverse learners, improve institutional effectiveness, and expand access to knowledge. Subcategories: Regular Education, Supplementary Education, Educational Institutions and Administration, Instructional Methods and Materials, Curriculum and Academic Structure, Educational Systems and Policy.

Continued on next page

Code	Title	# Subcategories	Description
EL	Ecology	5	The Ecology application domain includes software that analyzes relationships between organisms and their environments across terrestrial, aquatic, and atmospheric systems. It supports studies in ecosystem dynamics, biodiversity, and environmental change. Subfields include ecophysiology, which examines how environmental factors influence biological function; human ecology, which focuses on sustainable development and urban systems; and plant and animal ecology, which explores species interactions, habitats, and ecological balance. Ethology applications further extend this domain by modeling behavioral patterns such as migration and hibernation. These tools are essential for research, conservation, and environmental management. Subcategories: General Ecology, Ecophysiology, Human Ecology, Plant and Animal Ecology, Ethology.
EN	Energy	6	The Energy application domain includes software that supports the production, distribution, and management of energy resources. It covers energy policy analysis, consumption tracking, and economic evaluation, as well as the technical aspects of energy systems. Applications assist in managing fossil fuels, biomass, nuclear power, and renewable energy sources. The domain also includes tools for energy conversion, storage, and distribution, as well as specialized areas such as microelectronics and nuclear reactor technology. Subcategories: Energy, Energy Resources, Fuels, Energy Technology, Electronics Engineering, Nuclear Engineering.
FN	Finance	6	The Finance application domain includes software that manages and analyzes financial activities in both the public and private sectors. It supports government budgeting, taxation, and fiscal administration, as well as corporate finance operations such as credit, banking, and capital markets. Applications also handle budgeting tools, investment tracking, resource allocation, and financial planning. The domain encompasses financial administration, including interest rate management, debt oversight, and financial risk control, as well as accounting systems for performance evaluation and compliance. Subcategories: Public Finance, Private Finance, Financial Systems, Budgeting and Financial Instruments, Financial Management, Accounting.
FQ	Physics-Chemistry	18	The Physics-Chemistry application domain includes software that supports the study of the fundamental properties of matter and energy and their interactions. It spans key areas of physics such as mechanics, thermodynamics, electromagnetism, waves, radiation, and particle dynamics, along with central fields of chemistry including organic, inorganic, analytical, and physical chemistry. Applications also cover spectroscopy, metrology, surface and molecular physics, and physical-chemical processes. Subcategories: Particle Physics, Acoustics and Optics, Wave Theory, Metrology, Mechanics, Solid and Fluid Physics, Thermodynamics, Electronics, Magnetism and Electromagnetism, Surface and Dispersion Physics, Radiation, Spectroscopy, Molecular Physics, General Chemistry, Analytical and Polymer Chemistry, Physical Chemistry, Organic Chemistry, Inorganic Chemistry.
GC	Geography	10	The Geography application domain includes software that supports the study and representation of Earth's physical and human landscapes. It covers physical geography, which analyzes landforms, geomorphology, and natural processes, and human geography, which explores population patterns, economic systems, and political organization. The domain also includes regional and geographic orientation studies, as well as geospatial sciences such as geodesy, topography, and remote sensing. Specialized tools for photogrammetry, mapping, and cartographic methods facilitate accurate spatial analysis and visualization. Subcategories: Physical Geography, Human Geography, Regional Geography, Geographic Orientation, Geodesy, Topography, Photogrammetry, Mapping, Cartographic Methods, and Cartographic Plans.
GL	Geology	7	The Geology application domain includes software that supports the analysis of Earth's structure, composition, and geological processes. It covers surface dynamics like erosion and weathering, internal processes such as tectonics and volcanism, and structural geology including folds and faults. Applications extend to glaciology and marine geology, as well as historical geology fields like paleontology and stratigraphy. Economic geology tools assist in locating and analyzing mineral deposits, while geochemistry, geophysics, hydrogeology, and geotechnics provide insights into subsurface materials and conditions. Subcategories: Physical Geology, Glaciology, Geotectonics, Marine Geology, Historical Geology, Economic Geology, and Geochemistry/Geophysics/Geotechnics.
HB	Real Estate	2	The Real Estate application domain includes software that supports the analysis, planning, and management of housing systems and residential environments. It addresses housing markets, supply and demand dynamics, and public housing policies. Applications also classify and manage different housing types, including single- and multi-family units, student residences, elderly housing, and temporary or informal dwellings. Subcategories: Real Estate Policy and Markets, Real Estate Typology.
HD	Hydrology	4	The Hydrology application domain includes software that supports the analysis and monitoring of water in the environment. It covers the hydrological cycle, tracking the movement and distribution of water across rivers, lakes, and the atmosphere. Applications also support hydrography, which maps water bodies and flood zones, and hydrometry, which involves measuring rainfall, streamflow, evaporation, and sediment transport. The domain extends to oceanography, encompassing the physical, chemical, biological, and geological study of marine environments. Subcategories: Hydrology, Hydrography, Hydrometrics, Oceanography.

Continued on next page

Code	Title	# Subcategories	Description
IF	Information Management	10	The Information Management application domain includes software that organizes, processes, and distributes data and knowledge across various platforms and institutions. It supports the management of scientific, technical, and strategic information through systems for storage, retrieval, analysis, and dissemination. This domain encompasses libraries, archives, and documentation centers, as well as reprography tools like photocopying and microfilming. It also includes information systems and networks designed to optimize data flow, user access, and service delivery. Subcategories: Information, Information Documentation, Reprography, Information Materials, Library Management, Archival Science, Information Science, Information Services, User Information, Data Processing.
IN	Manufacturing	5	The Manufacturing application domain includes software that supports industrial production, transformation processes, and technological development. It covers a broad range of sectors, including metallurgy, electronics, chemicals, textiles, and food processing, as well as extractive industries like mining. Applications assist with industrial policy planning, production monitoring, and innovation management. The domain also integrates engineering tools for product design, testing, and manufacturing efficiency, alongside systems for managing technological research and the adoption of appropriate production technologies. Subcategories: Manufacturing, Industrial Technology, Industrial Engineering, Extractive Industry, Transformation Industry.
MA	Environment	4	The Environment application domain includes software that supports the protection, management, and improvement of natural and built environments. It addresses the development of environmental policies, conservation of renewable and non-renewable resources, and the monitoring and control of pollution—including air, water, soil, and noise pollution. Applications also focus on environmental quality, using data and engineering solutions to assess and improve living conditions. Subcategories: Environment, Natural Resources, Pollution, Environmental Quality.
ME	Meteorology	3	The Meteorology application domain includes software that supports the study and forecasting of atmospheric conditions. It encompasses atmospheric physics and dynamics, focusing on variables such as air pressure, wind, temperature, humidity, and solar radiation. Applications also support climatology, which analyzes long-term climate trends, seasonal patterns, and their influence on agriculture and ecosystems. Subcategories: Meteorology, Atmospheric Science, Climatology.
MT	Mathematics	6	The Mathematics application domain includes software that supports the development and application of mathematical theories and methods. It spans foundational areas such as logic, algebra, geometry, analysis, and calculus, which form the basis of formal reasoning and quantitative problem-solving. Applied mathematics extends these tools to real-world contexts through modeling, statistics, probability, and operations research. Subcategories: Logic, Algebra, Geometry, Analysis, Calculus, Applied Mathematics.
PD	Pedology	3	The Pedology application domain includes software that analyzes the formation, structure, and classification of soils in their natural settings. It addresses physical, chemical, mineralogical, and biological soil properties, supporting studies of soil horizons and development processes through morphopedological analysis. Applications help identify and map soil types, aiding in agricultural planning, land use management, and environmental conservation. Subcategories: Soil Science, Soil Formation, Soil Types.
PL	Public Policy	2	The Public Policy application domain includes software that supports the analysis and development of governmental policies and political systems. It draws on political science to model institutional structures, understand governance mechanisms, and evaluate public decision-making processes. Applications explore theories of power, state organization, political regimes, and policy implementation. Subcategories: Public Policy Science, Public Policy Systems.
PR	Social Security	3	The Social Security application domain includes software that supports the delivery and management of social welfare programs. It covers social security systems, including public and private retirement and pension schemes, as well as various forms of social assistance such as healthcare, housing, food aid, and rehabilitative services. These applications help implement policies, track benefits, and coordinate services aimed at reducing vulnerability and improving quality of life. They are essential for ensuring equitable access to support systems and strengthening the social safety net through both public and private initiatives. Subcategories: Social Security, Retirement Benefits, Social Assistance.
PS	Psychology	3	The Psychology application domain includes software that supports the study and application of mental processes and human behavior. It covers areas such as developmental, clinical, social, and educational psychology, as well as the analysis of sensory, cognitive, and emotional functions. Applications also explore behavioral theories, including behaviorism, existentialism, and reinforcement models. These tools are used in research, therapy, education, and mental health services to better understand psychological functioning and promote well-being, learning, and personal growth. Subcategories: Psychology, Human Behavior, Psychological Theories.
SD	Health	11	The Health application domain includes software that supports the delivery, management, and advancement of healthcare services. It covers public and mental health policy, healthcare administration, and disease prevention and treatment. Applications range from medical diagnostics and therapies to systems for hospital, ambulatory, and home-based care. The domain also includes general and specialized medicine, pharmacology, dentistry, and biomedical sciences. Subcategories: Public Health, Health Administration, Disease, Disabilities, Medical Assistance, Medical Diagnostics, General Medicine, Medical Specialties, Biomedical Engineering, Pharmacology, Dentistry.

Continued on next page

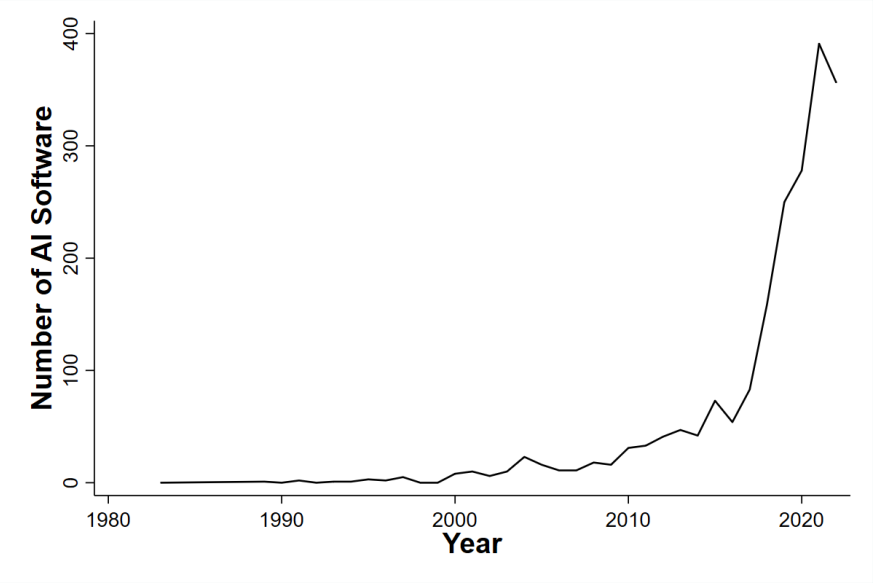
Code	Title	# Subcategories	Description
SM	Sanitation	5	The Sanitation application domain includes software that manages systems essential for public health and environmental quality. It encompasses sanitary engineering, basic sanitation services, and the treatment and disposal of various types of waste—solid, liquid, chemical, and industrial. Applications support public cleaning operations, including garbage collection and urban drainage, as well as water supply systems that handle the capture, treatment, and distribution of drinking water. The domain also covers sewage management, with tools for monitoring and treating both domestic and industrial wastewater. Subcategories: Sanitation Systems, Waste Management, Urban Cleaning, Water Supply, Wastewater and Sewage.
SV	Services	4	The Services application domain includes software that supports a wide array of public and private service activities essential to social and economic life. It encompasses utilities such as electricity, water, and telecommunications, as well as services related to maintenance, personal care, and public safety. Security applications manage insurance, risk assessment, and asset protection. Retail-focused tools handle trade operations, commercial logistics, and market policies, while tourism systems support travel planning, infrastructure management, and visitor services. Subcategories: Public and Private Services, Insurance, Commerce, Tourism.
TB	Human Resources	6	The Human Resources application domain includes software that manages the organization and development of labor and workforce systems. It covers different types of work—including manual, intellectual, rural, and technical—while addressing labor conditions, ergonomics, and employment policies. Applications support workforce planning, wage analysis, job classification, and labor market monitoring, as well as the structure of occupations through unions, councils, and professional associations. The domain also includes tools that promote worker well-being, such as leisure and recreation management. Subcategories: Organization of Work, Human Resources, Labor Market, Ergonomics, Occupational Structure and Professional Organizations, Leisure.
TC	Telecommunications	4	The Telecommunications application domain includes software that supports the planning, operation, and regulation of communication systems. It encompasses telecommunication policies and models, as well as technical systems such as telephony, television, radiocommunication, and data transmission. Engineering applications focus on the infrastructure for signal transmission and reception, including wired and wireless networks. The domain also covers the management of telecom services, network operations, and communication stations. Subcategories: Telecommunication Policy, Telecommunication Systems, Telecommunication Engineering, Network Services and Infrastructure.
TP	Logistics	5	The Logistics application domain includes software that coordinates the movement of goods and people across transport networks. It covers transport policy, infrastructure planning, and service operations across road, rail, air, and water modes. Applications support cargo and passenger management, vehicle routing, terminal operations, and multimodal integration. Logistics software also optimizes supply chains, enabling real-time tracking, route planning, and load management. Subcategories: Transport Policy and Planning, Transportation Networks and Infrastructure, Passenger and Freight Transport Services, Transport Engineering and Modal Systems, Modes of Transportation.
UB	Urban Planning	5	The Urban Planning application domain includes software that guides the development and organization of urban spaces. It supports land use planning, zoning, real estate regulation, and infrastructure design. Applications also address urban mobility, public services, and architectural projects across residential, commercial, and institutional buildings. Subcategories: Urban Planning and Design, Urban Land and Property, Urban Area Structure and Zoning, Urban Circulation and Infrastructure, Architecture.

Note: This table presents the code, title, number of subcategories, and description of each broad application domain. The description of each broad application domain was created by the authors using information from the original classification provided by the NIIP, Instituto Nacional da Propriedade Industrial (INPI) (2015b), and an inspection of the software in each classification.

A.3 Summary Statistics of the Software Dataset

This section presents summary statistics for the software dataset. Figure A7 displays the number of AI software registrations over time. Consistent with the trends discussed in Section 3, there is a sharp increase in AI software creation beginning around 2013.

Figure A7: **Number of New AI Software Registrations Over Time**

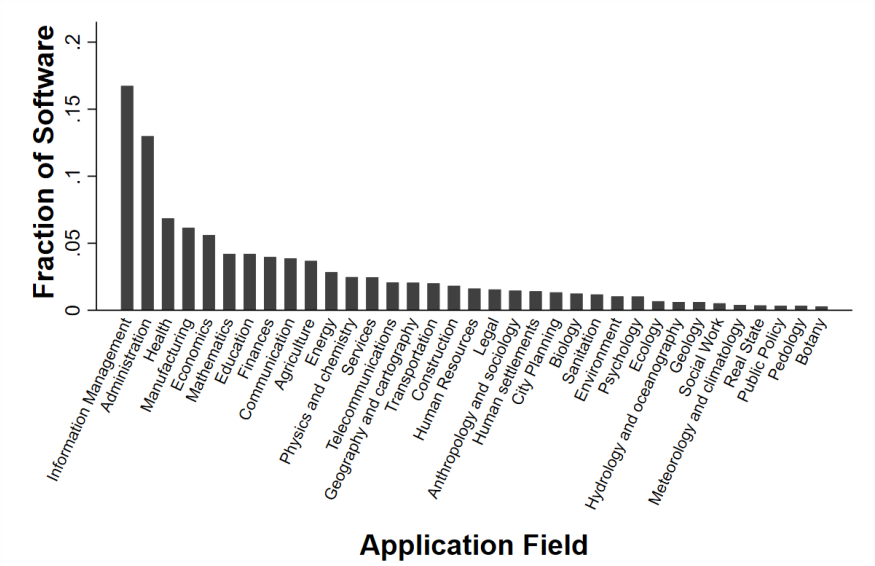


Notes: This figure plots the number of newly registered software classified as artificial intelligence (AI) in Brazil from 1987 to 2023.

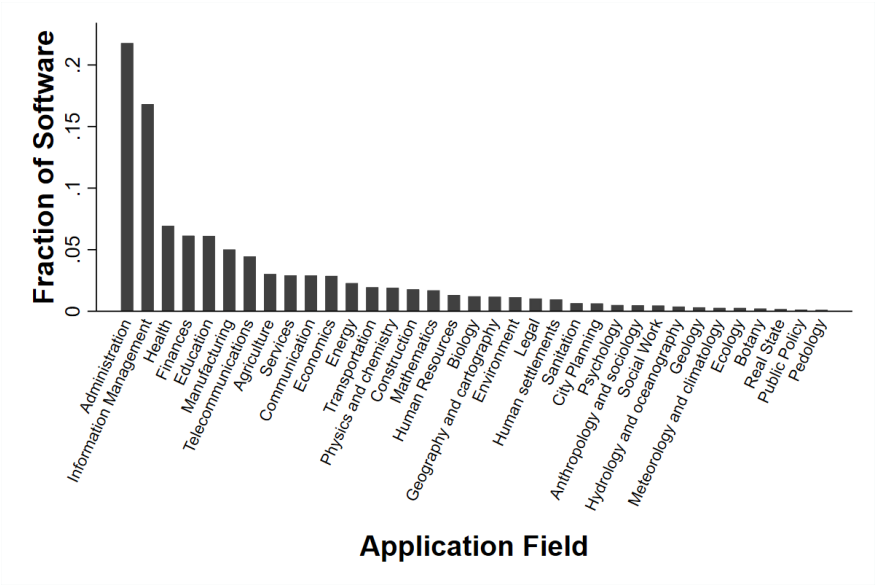
Figure A8 compares the distribution of application domains between AI and non-AI software. As shown in Figure A8b, non-AI software is more heavily concentrated in administrative and information management domains, whereas AI software is more broadly distributed across different areas.

Figure A8: **AI and Non-AI Application Domain**

(a) Fraction of Application Domain Among AI Software



(b) Fraction of Application Domain Among Non-AI Software

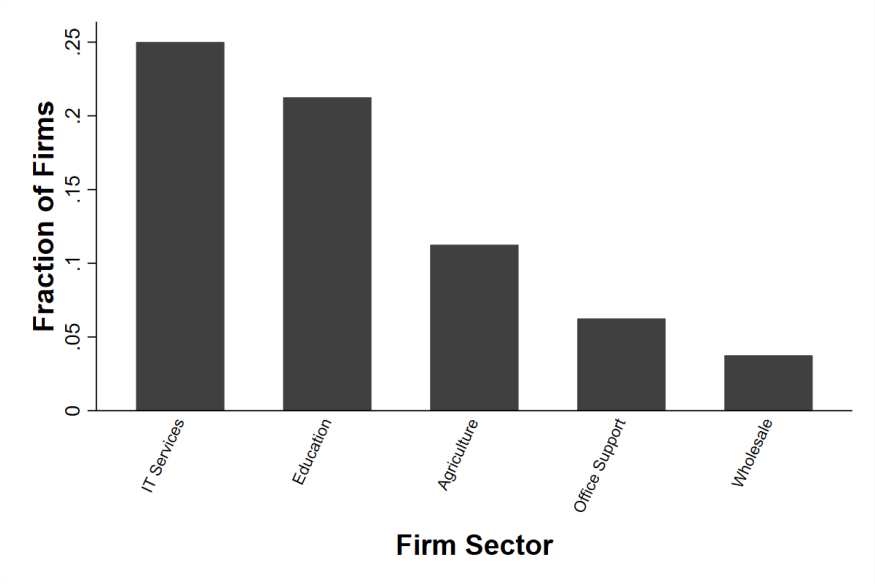


Notes: These figures present the distribution of AI software across different application domains, as defined by the NIIP. The application domain classification was developed by the NIIP to indicate the areas in which each software is intended to be used. Because a single software program may be assigned to multiple domains, each figure reports the fraction of each domain relative to the total number of domain classifications. Figure A8a plots the distribution of application domain among AI software while Figure A8b has the distribution of application domain among non-AI software.

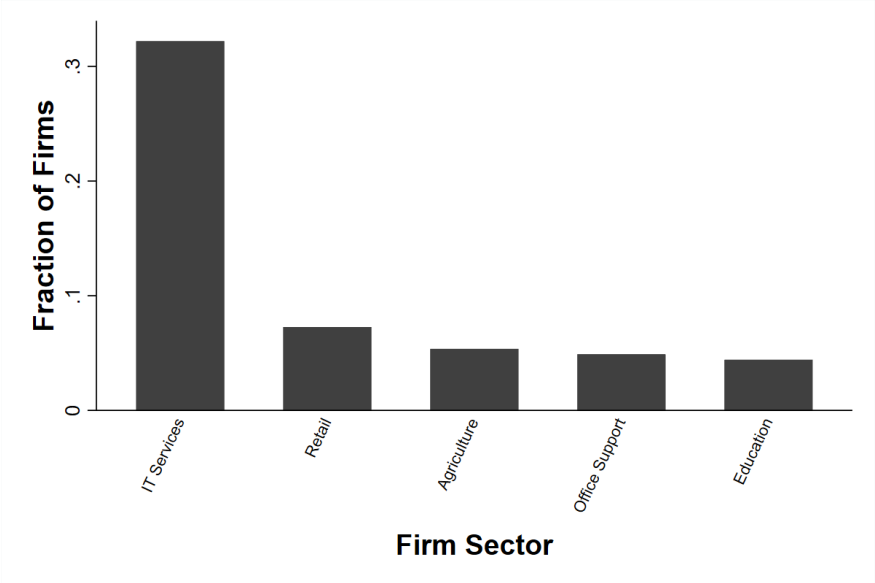
Figure A9 displays the distribution of AI and non-AI software across firm sectors. Among AI software, the most intensive sectors are Information Technology (IT) and Education, which includes universities and research institutions. In contrast, non-AI software is also concentrated in the IT sector but followed by the retail sector.

Figure A9: **Sectoral Distribution of Firms Creating AI and Non-AI Software**

(a) Sectoral Distribution of Firms Creating AI Software



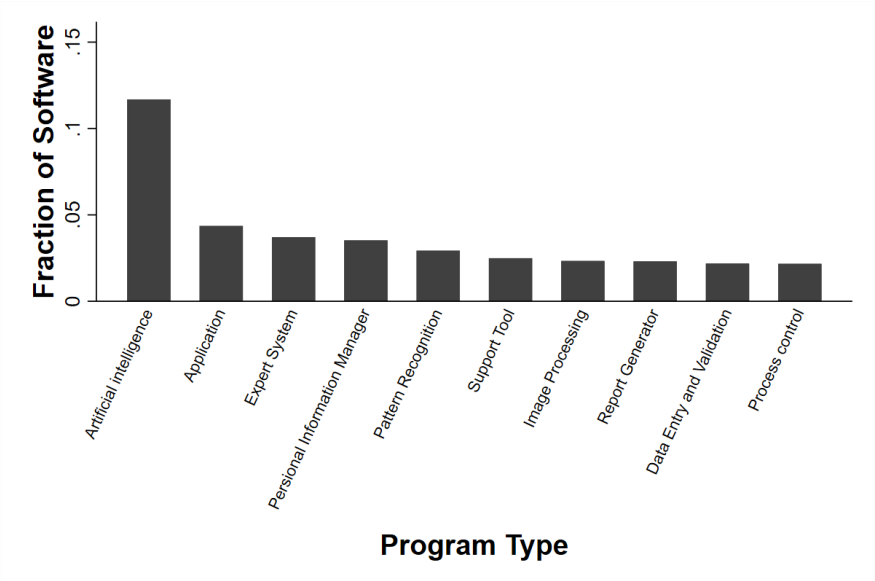
(b) Sectoral Distribution of Firms Creating non-AI Software



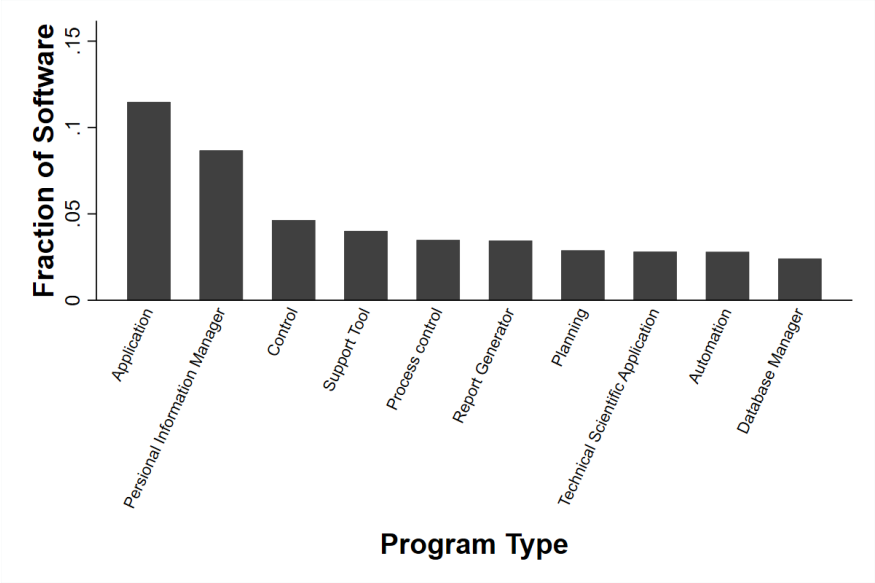
Notes: These figures show the distribution of firms registering AI and non-AI software across economic sectors.

Figure A10: **Fraction of Technical Class Among AI and Non-AI Software**

(a) Fraction of Technical Class Among AI Software



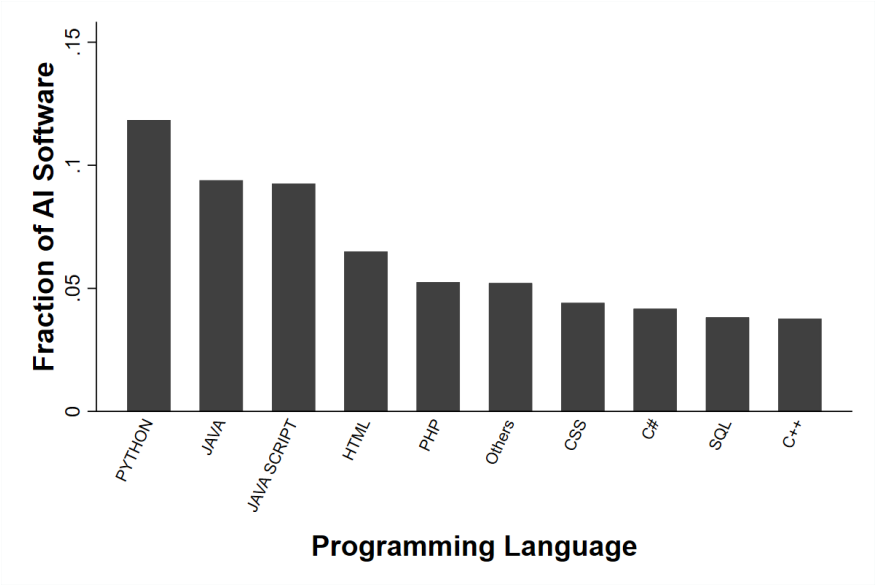
(b) Fraction of Technical Class Among non-AI Software



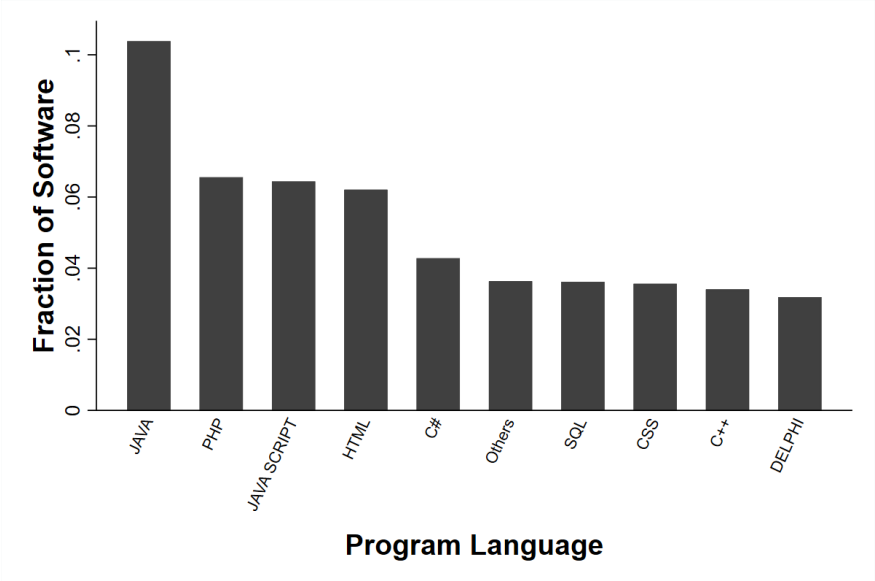
Notes: These figures show the distribution of technical class among AI and non-AI software.

Figure A11: **Fraction of Different Programming Languages Among AI and Non-AI Software**

(a) Fraction of Programming Language Among AI Software

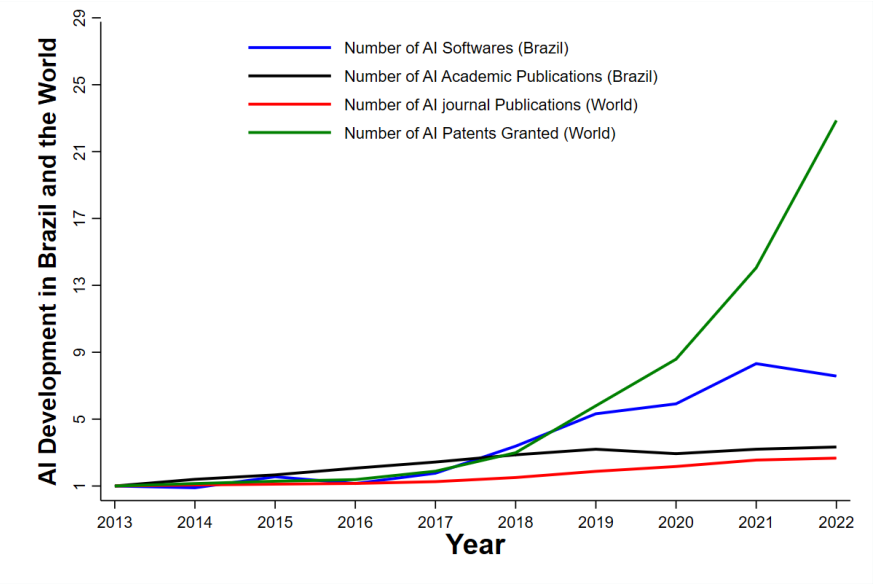


(b) Fraction of Programming Language Among non-AI Software



Notes: These figures show the distribution of programming languages among AI and non-AI software.

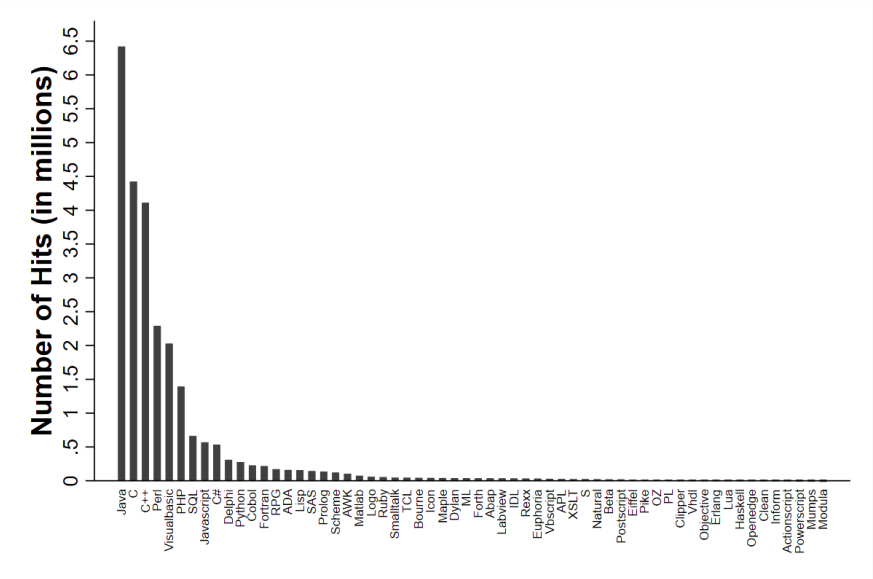
Figure A12: **AI Development in Brazil Follows Global Trend**



Notes: This figure plots the time series of four indicators of AI development: (i) the number of AI software programs registered in Brazil's NIIP database, (ii) the number of AI-related academic publications in Brazil, (iii) the number of AI-related journal articles globally, and (iv) the number of AI patents granted worldwide. Data for items (ii)–(iv) are from Maslej et al. (2025), while the software registration data are drawn from the NIIP registry.

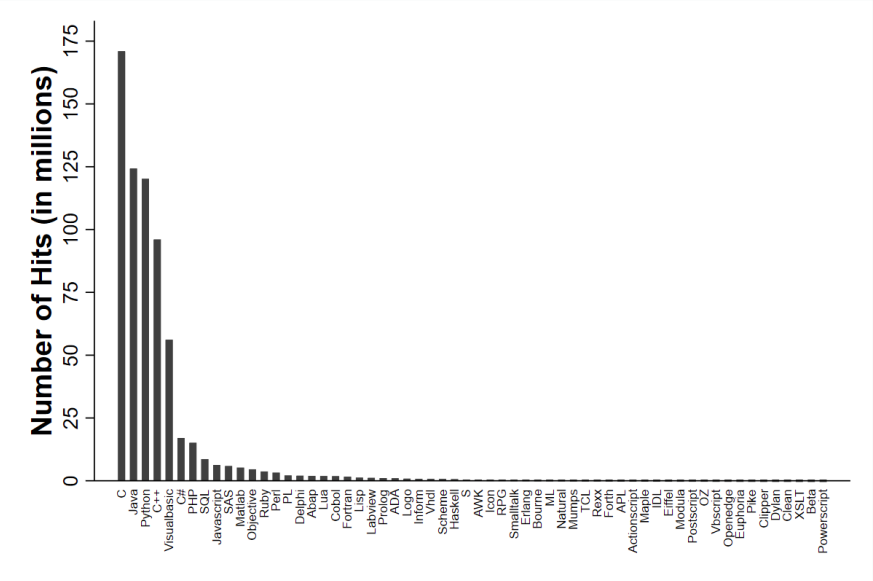
A.4 Summary Statistics of the TIOBE Index

Figure A13: Distribution of TIOBE index for programming languages in 2003



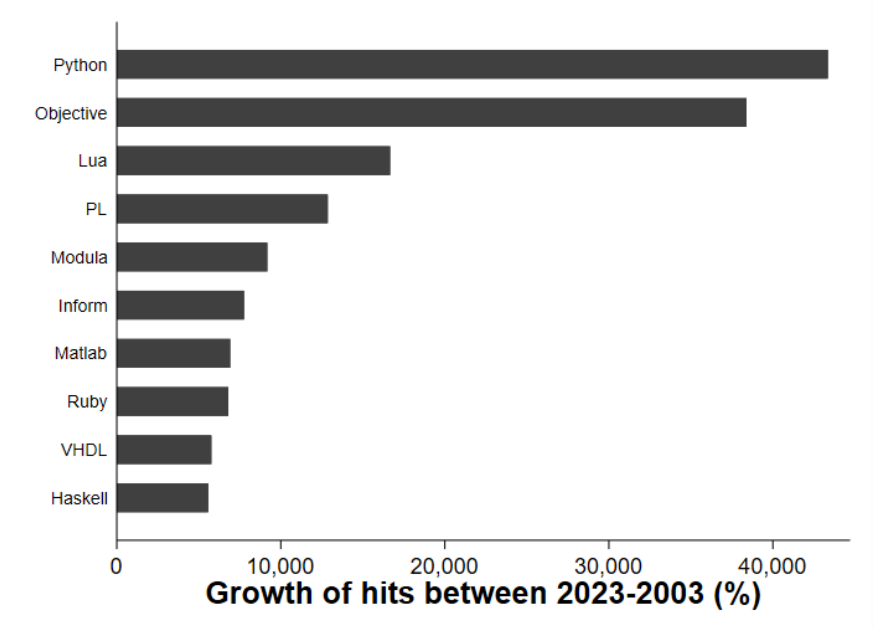
Notes: This figure shows the number of hits in Google.com in 2003 for the top programming languages using data from TIOBE.

Figure A14: Distribution of TIOBE index for programming languages in 2023



Notes: This figure shows the number of hits in Google.com in 2023 for the top programming languages using data from TIOBE.

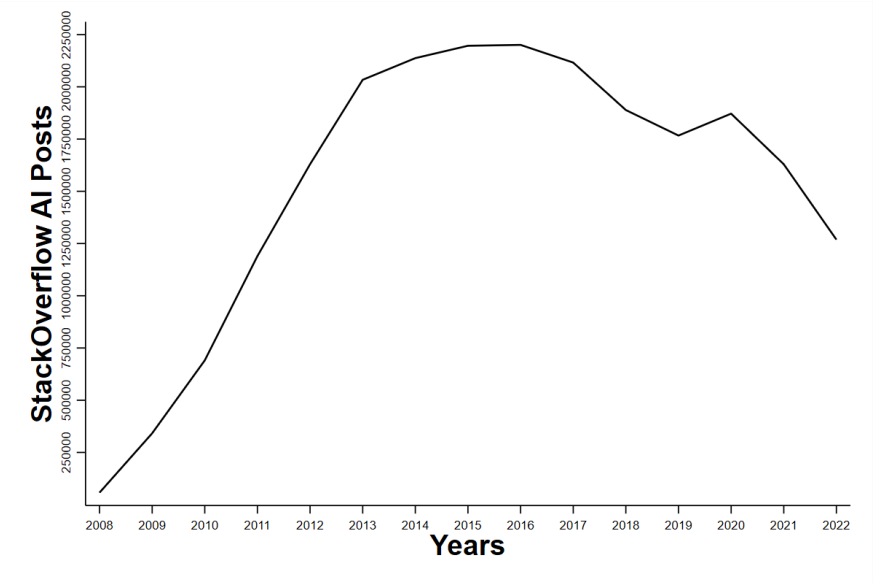
Figure A15: Top 10 programming languages with largest growth



Notes: This figure plots the programming languages with largest growth rate in number of hits in Google.com between 2003 and 2023 using data from TIOBE.

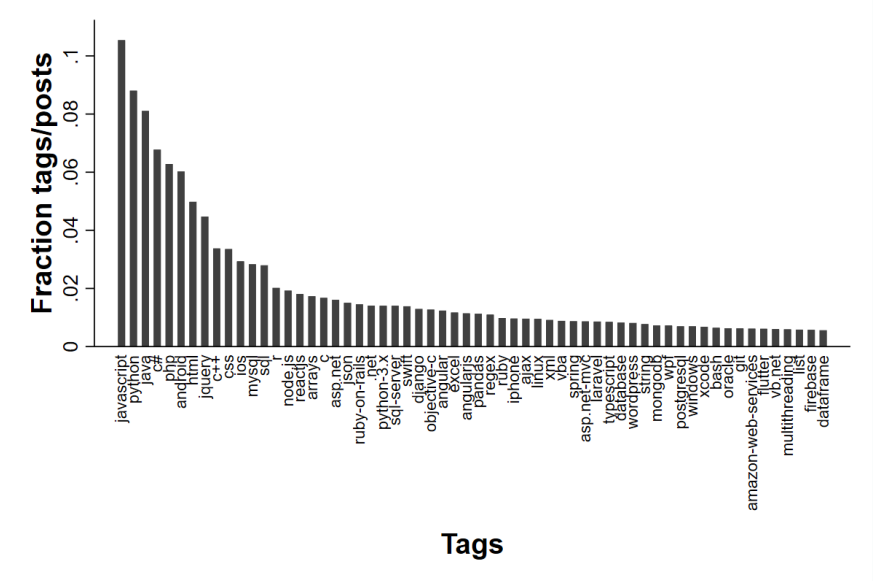
A.5 Statistics of Stack Overflow Posts

Figure A16: Number of stack overflow posts per year



Notes: This figure plots the number of new Stack Overflow posts over time.

Figure A17: Top 25 most popular tags



Notes: This figure plots the most common tags in StackOverflow and the share of posts with each one of them.

A.6 Specialization of Programming Languages

Table A5: **Main Programming Language for Each Technical Class**

Technical Class	**Main Language**	**Share**
Applications	javascript	.431
Planning	javascript	.494
Control	javascript	.534
Audit	javascript	.553
Accounting	javascript	.560
Automation	javascript	.440
Office Automation	javascript	.457
Commercial automation	javascript	.421
Banking Automation	javascript	.383
Industrial automation	javascript	.327
Process control	javascript	.541
Manufacturing Automation (CNC, Robotics)	javascript	.342
Automotive Electronics	c#	.419
Performance evaluation	javascript	.552
Resource Accounting	javascript	.587
Data communication	javascript	.517
Terminal Emulators	c#	.475
Teleprocessing Monitors	javascript	.442
Device and Peripheral Management	javascript	.447
Data Communication Network Manager	javascript	.450
Local network	javascript	.397
Telephone and Telegraph Switching	javascript	.500
Additional Functions Implementer	javascript	.475
Operation and Maintenance Manager	javascript	.510
Central Operation and Maintenance Terminal	javascript	.500
Systems Development Support Tools	javascript	.426
Application Generator	javascript	.469
Computer Aided Software Engineering	sql	.412
Applications Development System	javascript	.435
Routine Libraries (”Libraries”)	javascript	.363
Programming Support	javascript	.320
Documentation Support	javascript	.441
Systems Converter	javascript	.509

Continued on next page

Table A5 – continued from previous page

Technical Class	Main Language	Share
Entertainment	javascript	.384
Animated Games (”arcade games”)	c#	.548
Drawing Generators	c#	.450
Simulators for Leisure	c#	.457
Support Tool	javascript	.416
Word Processors	javascript	.500
Electronic Spreadsheets	javascript	.503
Graph Generators	javascript	.558
Information Manager	javascript	.463
Database Manager	javascript	.511
Screen Generator	javascript	.522
Report Generator	javascript	.536
Data dictionary	javascript	.578
Data Entry and Validation	javascript	.482
File Maintenance	javascript	.482
Data recovery	javascript	.401
Artificial intelligence	python	.328
Expert Systems	javascript	.321
Natural Language Processing Systems	python	.490
Instrumentation	c#	.321
Test and Measurement Instrumentation	javascript	.405
Biomedical Instrumentation	javascript	.321
Analytical Instrumentation	javascript	.532
Languages	javascript	.528
Compiler	javascript	.544
Assembler	java	.519
Pre-Compiler	javascript	.500
Cross Compiler	javascript	.519
Pre-Processor	java	.538
Interpreter	java	.421
Procedural Language	javascript	.500
Non-Procedural Language	javascript	.564
Security and Data Protection	javascript	.476
Password	javascript	.487
Cryptography	javascript	.519

Continued on next page

Table A5 – continued from previous page

Technical Class	Main Language	Share
Maintaining Data Integrity	javascript	.546
Access Control	javascript	.528
Simulation and Modeling	python	.297
Flight/Car/Submarine Simulator	c#	.485
Operating Environment Simulators	javascript	.417
CAE/CAD/CAM/CAL/CBT	c#	.419
Operational system	javascript	.375
Input and Output Interface	javascript	.471
Basic Disk Interface	python	.389
Communication Interface	javascript	.409
User Manager	javascript	.570
Device Administrator	javascript	.473
Process Controller	javascript	.457
Network Controller	sql	.365
Command Processor	java	.371
Technical-Scientific Applications	python	.365
Operational Research	python	.359
Pattern Recognition	python	.439
Image Processing	python	.427
Teleinformatics	javascript	.364
Terminals	sql	.450
Data Transmission	javascript	.479
Data Switching	javascript	.473
Utilities	javascript	.361
Data Compressor	javascript	.423
Storage Media Converter	javascript	.462
Classifier/Interleaver	python	.426
Spool Controller	c#	.444
File transference	javascript	.553

B Empirics

B.1 Text Similarity

In this section, I describe in detail how I calculate the text similarity between software description and occupational tasks.

Parsing. I transform documents into a vector using words as tokens, i.e., 1-gram.

Selection. To avoid counting frequent and uninformative words, such as the Portuguese equivalent for "the" and "and", I drop terms that appear in more than 80% of documents.

Vectorization. Following the previous steps, I characterize each document with a vector of dummies for words it contains. Let $m \in \{1, ..., M\} = \mathcal{M}$ be the set for words in the documents. Let c_{km} be a dummy variable taking the value of 1 if the document k contains word m. Therefore, document k can be represented by vector c_k with entries c_{km}.

Normalization. Rare words are more important for characterizing differences across documents than common words. To take that into account, I weight each word using term-frequency-inverse-document-frequency (tf-idf). Each term m of the dataset is weighted by:

$$\omega_m = \log\left(\frac{K+1}{d_m+1}\right) + 1 \text{ where } d_m = \sum_k \mathbb{I}\left\{c_{km} > 0\right\}.$$

After weighting, each document is represented by word frequency vector f_k with entries:

$$f_{km} = \frac{\omega_m c_{km}}{\sqrt{\sum_{m'} (\omega_m c_{km})^2}}.$$

Singular Value Decomposition. Let F represent the entire corpus. Latent semantic analysis decomposes the matrix F as

$$F = U\Sigma V'$$

where U is the term-factor matrix, Σ the diagonal matrix of singular values, and V' the factor-document matrix.

Rank-k approximation. I set to zero all the singular values outside the top 300, following Schwarz (2019). Using the decomposition, I recover the matrix $\tilde{F}$ given by

$$\tilde{F} = U\tilde{\Sigma}V'$$

where $\tilde{\Sigma}$ has only the first 300 singular values different from zero.

Similarity Scores. The similarity between software s and tasks of occupation o is given by:

$$s_{so} = \sum_{m \in \mathbb{M}} \tilde{f}_{sm} \times \tilde{f}_{om}. \tag{8}$$

B.2 Summary Statistics of AI Exposure

Figure A18 shows average AI exposure in 2022 across 1-digit occupation groups. Brackets indicate the range from the bottom to the top decile within each category. Exposure varies widely both across and within groups: managers, science professionals, and administrative workers are the most exposed, while agricultural workers are the least exposed.[47] Within-group variation is particularly notable: the most exposed manufacturing worker, for example, faces greater AI exposure than the average administrative worker. This wide distribution of AI exposure across occupations likely reflects its diverse range of applications, as discussed in Section 3.

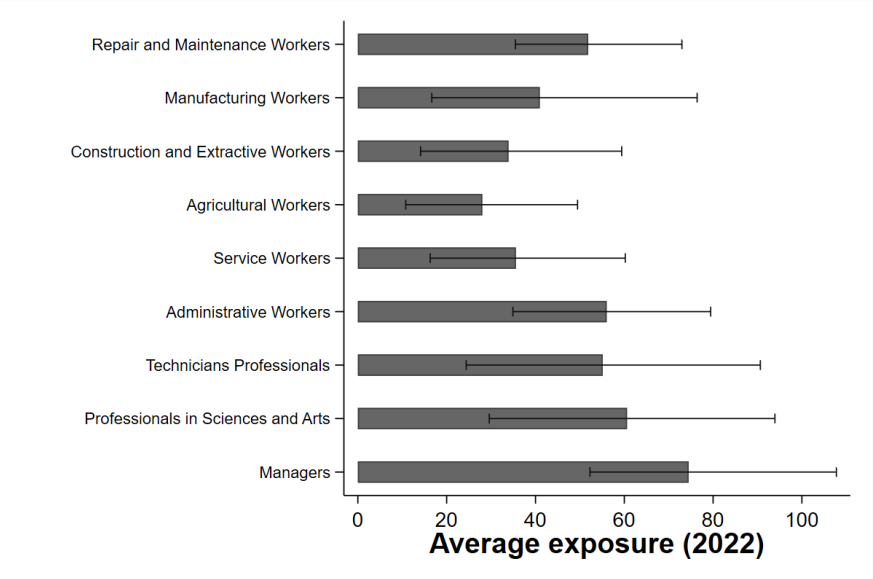
Table A6, in the Appendix, shows the correlation between the 2003 characteristics of each occupation and their exposure to AI software in 2022. Occupations more exposed to AI tend to have higher hourly wages, greater education, and larger workforces. They are also more likely to involve tasks related to planning, management, and interpersonal activities, which is consistent with the finding that 40% of AI software targets administrative tasks.

B.3 Comparison to existing exposure metrics

Table A7 shows that the exposure measure defined in Equation (1) is positively correlated ($\rho = 0.417$) with the GPT-4-based task replaceability scores in Gmyrek et al. (2023), but

[47] Prytkova et al. (2024), Gmyrek et al. (2023), Felten et al. (2018), and Webb (2020) also find the same general patterns. In the Appendix Section B.3, I compare the AI exposure measure 1 to other measures developed in the literature.

Figure A18: **Large Variance within and Across Broad Occupations in AI Exposure**



Notes: This figure plots the average AI exposure in 2022 at a one-digit CBO occupational classification. The bars plot the average AI exposure and the lines the top and bottom decile.

negatively correlated with both the feasibility scores of Frey and Osborne (2017) and the patent-similarity measure of Webb (2020). One possible explanation is that earlier metrics attempt to predict what AI could do—drawing on expert judgment, capability benchmarks, or patent abstracts—while the NIIP-based measure captures what AI is already doing in practice.

The low correlations across metrics highlight the lack of consensus in the literature on which occupations are most exposed to AI. The two most used metrics of AI exposure, those developed by Webb (2020) and Frey and Osborne (2017), are negatively correlated while the remaining pairwise correlation does not exceed 0.417.

Table A6: **Correlation Between AI Exposure and Characteristics of the Occupation**

	(1) $AI\ Exposure_{o,2022}$	(2) $AI\ Exposure_{o,2022}$	(3) $AI\ Exposure_{o,2022}$
$log(hour\ wage)_{o,2003}$	0.651^{***}	0.556^{***}	0.483^{***}
	(0.0555)	(0.0599)	(0.0627)
$log(yrs.\ educ.)_{o,2003}$	1.130^{***}	1.345^{***}	1.091^{***}
	(0.164)	(0.205)	(0.214)
$log(n.\ workers)_{o,2003}$	0.0531^{***}	0.0629^{***}	0.0735^{***}
	(0.0125)	(0.0130)	(0.0136)
$\mathbb{I}\{use\ computer\}_o$	0.162^{**}	0.241^{***}	0.222^{***}
	(0.0630)	(0.0650)	(0.0665)
$\mathbb{I}\{operate\ machine\}_o$	-0.0601	-0.0291	-0.0116
	(0.0626)	(0.0675)	(0.0676)
$\mathbb{I}\{Planning\ or\ Execution\}_o$	0.295^{***}	0.207^{***}	-0.00117
	(0.0757)	(0.0753)	(0.0762)
$\mathbb{I}\{Creation\ and\ Innovation\}_o$	-0.255^{**}	-0.249^{**}	-0.176^{**}
	(0.0837)	(0.0831)	(0.0823)
$\mathbb{I}\{Management\}_o$	0.333^{***}	0.282^{***}	0.217^{**}
	(0.0599)	(0.0600)	(0.0619)
$\mathbb{I}\{Interpersonal\}_o$	0.381^{***}	0.435^{***}	0.432^{***}
	(0.0635)	(0.0688)	(0.0726)
Controls		1-digit Occ. FE	2-digit Occ. FE
Observations	2,148	2,148	2,148
R^2	0.3809	0.4100	0.4731

Notes: This table reports regression results where the dependent variable is the stock cosine similarity in 2022. Each regressor is defined at the occupational level in 2003: *log(hour wage)* is the log of average hourly wage; *log(yrs. educ.)* is the log of average years of schooling; *log(n. workers)* is the log of the number of workers; $\mathbb{I}\{use\ computer\}$ is a dummy if among the tools used by the occupation is a computer; $\mathbb{I}\{operate\ machine\}$ is a dummy taking one if among the tasks performed by the occupation there is machine operation; $\mathbb{I}\{Planning\ or\ Execution\}$ is a dummy taking one if the occupation performs tasks related to planning or execution, $\mathbb{I}\{Creation\ and\ Innovation\}$ is a dummy taking one if the occupation performs tasks related to creation or innovation, $\mathbb{I}\{Management\}$ is a dummy if the occupation performs tasks related to management, and $\mathbb{I}\{Interpersonal\}$ is a dummy taking one if the occupation performs tasks related to interpersonal relationships. Standard errors in parentheses.Standard errors are in parentheses. * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$.

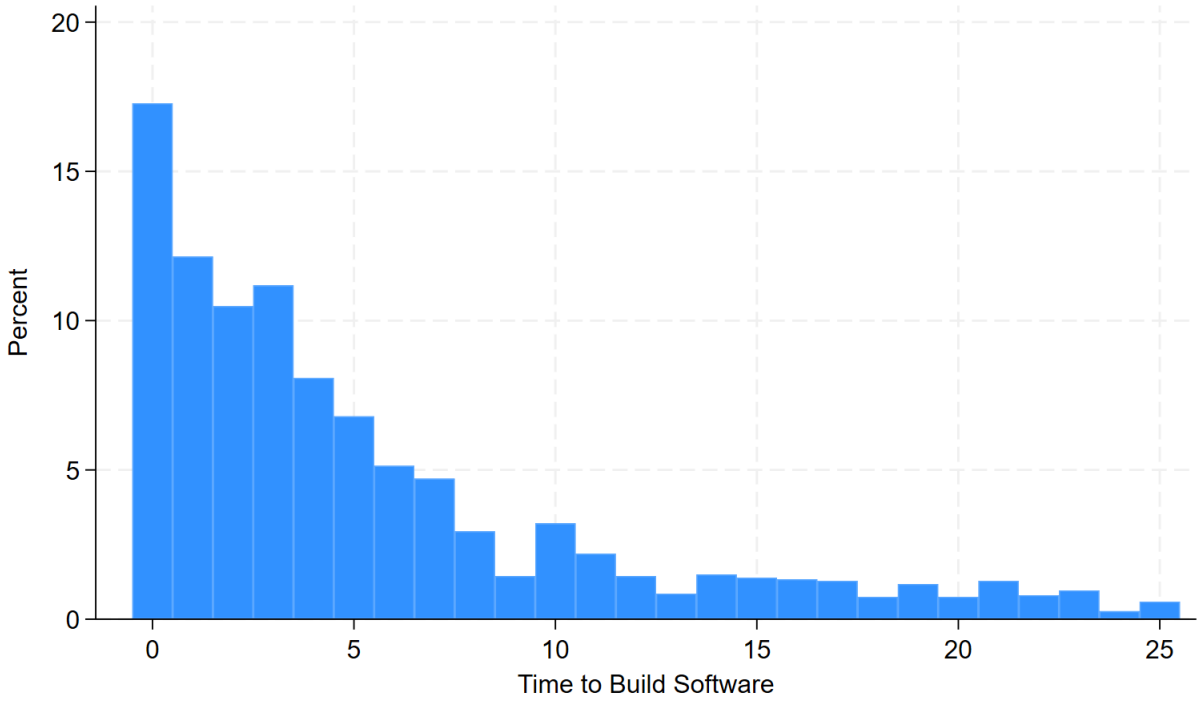
Table A7: **Correlation Between Different AI Exposure Metrics**

Variables	(1)	(2)	(3)	(4)	(5)	(6)
(1) $AI\ Exposure_{o,2022}$	1.000					
(3) Felten et al. (2018)	0.198	1.000				
(4) Frey and Osborne (2017)	-0.338	-0.296	1.000			
(5) Gmyrek et al. (2023)	0.417	0.038	-0.188	1.000		
(6) Prytkova et al. (2024)	0.147	0.064	0.132	0.249	1.000	
(7) Webb (2020)	-0.178	0.189	0.167	-0.143	0.120	1.000

B.4 Software Development Time

To estimate the time required to develop software, I measure the interval between when a programmer is hired and when they are first credited as an author of a registered software. To construct this measure, I merge NIIP software registration records with RAIS data, linking each software entry to the firm that owns it and to the programmer who authored it. This produces a firm–worker panel from 2002 to 2016, restricted to software programmers. For each programmer–firm pair, I identify the initial hiring date, keeping only the first recorded employment spell to avoid ambiguity from repeated hires. The development time is then calculated as the number of days between that hiring date and the publication of the worker's first registered software.

Figure A19: **Distribution of Time to Build Software in Years**



B.5 Mincer Regression to Identify Abilities

To estimate individual's ability, I run a Mincer regression using the version of RAIS with individual and firm identifiers. The dataset runs from 2003 to 2016, capturing only the initial

Table A8: **Time to Build Software**

	Years
Mean	5.4
Median	3
10th Percentile	0
25th Percentile	1
75th Percentile	7
90th Percentile	15

Notes: This table summarizes the distribution of time between hiring and the first software publication by a programmer. Time is measured in years.

years of the AI boom. The specification is given by

$$log(Wages_{i,t}) = \mu_i + X_{i,t} + \epsilon_{i,t}$$

where $Wages_{i,t}$ is monthly wages and μ_i is a worker fixed effect. $X_{i,t}$ is a set of controls including a 6-digit occupation fixed effect, an education-year fixed effect, an age fixed effect, a gender fixed effect, a 4-digit sector-year fixed effect, and a city-year fixed effect.

To calculate the average ability in occupation o in year t, used in Table 5, I average the fixed effect μ_i for workers in occupation o in year t. The wage residual in occupation o and year t is given by the average of $\epsilon_{i,t}$ for workers in that occupation in that year.

B.6 Relative Information Use Intensity

This appendix describes the construction of the occupation-level relative information use intensity, which measures the extent to which an occupation's tasks rely on using information to guide decisions and actions, as opposed to gathering or processing information. To construct the index, I use ChatGPT 5.2 to identify tasks that fall into two groups: tasks related to information processing and recordkeeping and tasks that benefit from access to information to make decisions or take action.

Information processing and recordkeeping. This category includes tasks whose core function is to gather, register, verify, code/classify, update, route, or report organizational information and documents. These tasks primarily maintain information systems rather than acting on information. Examples include filling out forms, updating registries or customer

records, entering transactions, checking documentation, archiving files, routing correspondence, processing routine invoices or payments, and maintaining inventory records.

Information use to make decisions or take action. This category includes tasks that benefit from access to information in order to take action, make decisions, coordinate activities, adjust processes, diagnose conditions, or control operations. Examples include adjusting operating parameters based on readings, responding to alarms, prioritizing maintenance interventions based on alerts or failure histories, releasing or retaining lots based on inspection results, rescheduling production or routes based on constraints and status updates, and operating equipment based on technical parameters.

Relative information use intensity. Let s_o^{use} and $s_o^{process}$ be the share of tasks of occupation o classified as information use and information processing, respectively. The relative information use intensity is given by

$$\textit{Relative Information Use Intensity}_o = \frac{s_o^{use}}{s_o^{use} + s_o^{process}}.$$

To guarantee that every occupation has a valid relative information use intensity, I use broad occupations to calculate this measure.

Prompts. To classify tasks into the two categories, I build two prompts by layering several prompt-engineering methods. First, I assign the model a domain role ("You are an occupational analyst"), a role-prompting approach shown to increase precision (Kong et al. 2024). Next, I describe the task context and objective and impose strict constraints, following instruction-style directive prompting with explicit output-control rules (Mao et al. 2025). I then define "information use"/"information processing," which improves LLM classification accuracy (Kumar et al. 2023, Hansen et al. 2024, Ruckdeschel 2025). That definition includes inclusion rules drawn from annotation-guideline prompting to reduce ambiguity and improve labeling consistency (Halterman and Keith 2025, Fonseca and Cohen 2024, Shinohara and Kawazoe 2026). I also add exclusion rules through contrastive prompting to tighten the scope of the task (Gao and Das 2024). The prompt lists example tasks associated with "information use"/"information processing," using few-shot prompting to improve accuracy (Brown et al. 2020, Min et al. 2022). Finally, I prompt the LLM to copy output verbatim from a file, a constraint shown to reduce hallucinations (Long et al. 2025, Reynolds and McDonell

2021). Because the task descriptions are in Portuguese, I used a prompt in Portuguese to identify the tasks related to information processing or use.

The translated prompt for identifying tasks that benefit from information use is given by:

> *You are an occupational analyst. You will receive (as an attachment) a CSV file containing the occupational profiles from CBO 2002. This file contains, for each occupation, a list of activities/tasks (one per row). Your job is to select and return ONLY CBO tasks that represent "information use" at work.*
> *Objective: Generate a single, unique list (do not group by occupation) of tasks whose content describes actions/decisions/adjustments/control/coordination that benefit from access to already-available information (generated by systems, sensors, reports, indicators, work orders, histories, forecasts, alerts, checklists, test/inspection results, etc.).*
> *How "information use" can appear across sectors: - Industry/processes: adjust variables/setpoints; define operating sequence; respond to alarms; stop/resume equipment; stabilize a process based on readings/limits. - Quality: release/hold batches; trigger containment/correction; change inspection parameters based on test/measurement results. - Maintenance: prioritize interventions; diagnose failures; decide replacement/repair; schedule shutdowns based on symptoms, alarms, history, and instruments. - Logistics/operations: prioritize queues; reroute routes/loads; reschedule service based on status, constraints, and alerts. - Agriculture/environment: define application/management/irrigation/crop practices based on soil/plant conditions, weather, pests/diseases, and productivity. - Construction/extractive: decide shoring/interdiction/continuation; adjust execution and safety based on tests, ground/face conditions, measurements, and inspections.*
> *Core OUTPUT rule (very important): - The output must contain ONLY tasks related to information use. - Tasks do NOT need to be "centered on interpreting information". They may also be tasks of action/decision/control/coordination/adjustment that benefit from access to information (even if the task text does not explicitly say "interpret").*
> *Examples of task types that QUALIFY (depending on the exact CBO text): plan, schedule, control, supervise, decide, prioritize, authorize, adjust/regulate, operate based on parameters, diagnose, intervene, trigger/activate, respond to alerts, release/hold, instruct, coordinate, replan, redefine routes/sequence, apply treatment/management, configure equipment.*
> *Exclusion rule (very important): - DO NOT include tasks whose core is collecting/recording/processing information to feed systems. Exclude tasks such as (depending on the exact CBO text): collecting data/samples; recording/noting; filling forms; typing/entering; classifying/coding for registration; filing/protocoling; archiving; organizing documents; checking documents for registration; "preparing/issuing reports" when it is routine recordkeeping/documentation (without decision/action). - Also DO NOT include items of "personal competencies" (e.g., "Demonstrate ..."), because they are not tasks. - Remove identical duplicates: if the same task appears in multiple occupations, list it only once.*
> *Source of tasks (mandatory): - Tasks must come EXCLUSIVELY from the attached file (CBO occupational profile). - Copy and paste task text exactly as it appears in the file (same spelling, accents/diacritics, capitalization, punctuation). - DO NOT paraphrase, DO NOT summarize, DO NOT "improve" the Portuguese, and DO NOT invent tasks.*
> *Output format (TXT-friendly; no extra comments): Return ONLY a bulleted list, one per line: - NOME_ATIVIDADE exactly as in the file - NOME_ATIVIDADE exactly as in the file - ...*
> *Final constraints: - Return ONLY the list (no introduction, no conclusion, no explanations of criteria). - Do not include occupation names, codes, or any other CSV fields — only the tasks. - Do not include tasks outside the file. - If you are unsure whether a task is "information use" or "recording/processing", be conservative and EXCLUDE it.*
>
> *OUTPUT EXAMPLES (only to illustrate format; DO NOT copy these examples in your final response): - Adjust process variables - Release batches according to inspection results - Prioritize maintenance interventions according to alarms and failure history - Reschedule routes/services according to constraints and operational alerts - Decide to interrupt operation under risky*

conditions

The translated prompt for identifying tasks related to information processing is given by:

You are an occupational analyst. You will receive (as an attachment) a CSV file containing the occupational profiles from CBO 2002. This file contains, for each occupation, a list of activities/tasks (one per line). Your job is to select and return ONLY CBO tasks related to collecting, generating, recording, verifying, coding/classifying, updating, routing/forwarding, or reporting information/data within firms.
Objective: Generate a single, unique list (do not group by occupation) of tasks whose content describes the "upstream handling" of information: capturing/recording/checking/organizing/labeling/updating/routing/reporting organizational data and documents (documents, records, transactions, registries, schedules, inventories) that sustain internal routines.
Selection rules (apply without stating them): - Select tasks typical of office routines and operational support that keep the organization running through documentation and recordkeeping. - Emphasize tasks related to: forms and records; invoices, billing, and payments; inventory and materials controls; schedules and timekeeping/time logs; purchase orders and follow-up; customer/order records; compliance and documentation; filing/archives and correspondence. - Include tasks from adjacent areas only when they are clearly routine information processing (e.g., capturing/recording orders, basic financial processing, documentary logistics support).
Exclusion rule (very important): - DO NOT include tasks whose core is USING information to make decisions, control/adjust production processes, supervise via indicators, optimize strategy, diagnose systems, make forecasts, or decide actions based on analyses. Examples of what to exclude (depending on the exact CBO wording): "define strategies," "make managerial decisions," "monitor KPIs for correction," "adjust process parameters," "operate equipment based on readings," "plan production," "supervise operations," "evaluate results for decisions." - Focus strictly on "upstream" handling (input/recording/validation/classification/filing/routing/reporting) and not the "downstream" use of information.
Source of tasks (mandatory): - Tasks MUST come EXCLUSIVELY from the attached file (CBO occupational profile). - Copy and paste the task text exactly as it appears in the file (same spelling, accents/diacritics, capitalization, and punctuation). - DO NOT paraphrase, DO NOT summarize, DO NOT "improve" the Portuguese, and DO NOT invent tasks.
Additional cleaning rules: - DO NOT include "personal competencies" items (e.g., "Demonstrate ..."), as they are not tasks. - Remove identical duplicates: if the same task appears in multiple occupations, list it only once. - If you are unsure whether a task is "processing/recordkeeping" or "use for decision," be conservative and EXCLUDE it.
Coverage: - Return a broad but manageable list: target 120–250 unique tasks (after deduplication). - Ensure diversity across task types covering routine finance/accounting, routine HR/payroll, purchasing, documentary logistics, administrative/customer registration, documentation/filing, and general office routines.
Practical CSV note: - The CSV delimiter is ";". The field `NOME_ATIVIDADE` *may contain ";" inside the text. - When splitting fields, treat only the first 8 occurrences of ";" as delimiters; the remainder belongs to* `NOME_ATIVIDADE`*.*
Output format (TXT-compatible; no extra comments): - Return ONLY a bulleted list, one per line: - `<NOME_ATIVIDADE exactly as in the file>` *-* `<NOME_ATIVIDADE exactly as in the file>` *- ...*
Final constraints: - The output must NOT list occupations, codes, descriptions, or any other CSV fields — only the tasks. - Do not include tasks outside the file. - Do not include headers, introductions, conclusions, or explanations.

OUTPUT EXAMPLES (only to illustrate the format; DO NOT copy these examples in your final response): - Check documents - File documents - Update registries - Fill out forms - Forward correspondence

B.7 Other Results

Table A9: **Effects of AI Between 1995–2002**

	(1) *log(Employment)*	(2) *log(Age)*	(3) *log(Yrs. Education)*	(4) *log(Tenure)*	(5) *log(Wage)*
AI Exposure	−0.0364	0.000501	−0.0210*	−0.0181	−0.0362
	(0.0486)	(0.00707)	(0.0121)	(0.0365)	(0.0275)
N	5,465	5,465	5,465	5,465	5,465

Notes: This table reports estimates of the effect of AI on labor market outcomes, following the specification in equation (2), restricting the sample to the period 1995–2002. Column 1 reports the effect of AI on the log of employment; column 2 on the log of average age; column 3 on the log of average years of education; column 4 on the log of average tenure at the current firm (in years); and column 5 on the log of average wage. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10,^{**}p < 0.05,^{***}p < 0.01$.

Table A10: **Programming-language popularity and software development**

	(1) *N. AI Software using language l*	(2) *N. AI Software using language l*	(3) *Shr. AI Software using language l*	(4) *Shr. AI Software using language l*
$\frac{\textit{hits language } l \textit{ in year } t}{\textit{hits all languages in year } t}$	88.59***		0.358***	
	(7.434)		(0.0239)	
$\frac{\textit{hits language } l \textit{ in year } t-2}{\textit{hits all languages in year } t-2}$		38.17***		0.395***
		(8.405)		(0.0238)
N	6 027	6 027	5 740	5 740

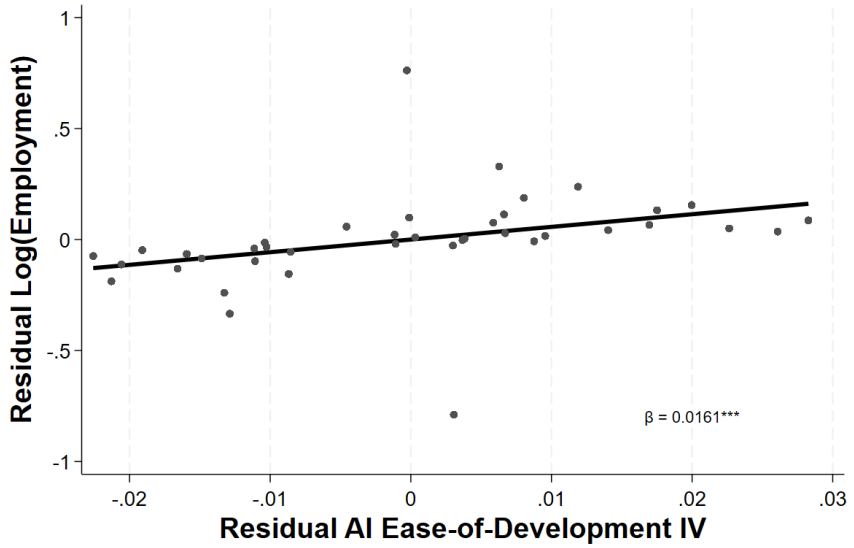
Notes: This table reports coefficients from an OLS regression relating each programming language's relative popularity in the United States to the number of AI software applications created in Brazil using that language. Popularity is the share of Google search hits for language l relative to all languages, measured contemporaneously (columns 1 and 3) or with a two-year lag (columns 2 and 4). Standard errors clustered by language appear in parentheses. $^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$.

Table A11: **Specialization in Stack Overflow and Specialization in the NIIP Dataset**

	(1) $Spec._{l,p}^{StackOf}$	(2) $Spec._{l,p}^{StackOf}$	(3) $Spec._{l,p}^{StackOf}$	(4) $Spec._{l,p}^{StackOf}$
$Spec._{l,p}^{NIIP}$	0.182***	0.186***	0.00556***	0.00566***
	(0.00440)	(0.00444)	(0.00168)	(0.00170)
N	49,980	49,980	49,980	49,980
R^2	0.033	0.034	0.871	0.871
	$log(Spec._{l,p}^{StackOf})$	$log(Spec._{l,p}^{StackOf})$	$log(Spec._{l,p}^{StackOf})$	$log(Spec._{l,p}^{StackOf})$
$log(Spec._{l,p}^{NIIP})$	0.386***	0.374***	0.0649***	0.0843***
	(0.0213)	(0.0226)	(0.0159)	(0.0157)
N	701	700	685	684
R^2	0.320	0.384	0.901	0.917
Fixed Effects:				
Programming Language	-	✓	-	✓
Technical Class	-	-	✓	✓

Notes: This table shows that the specialization of programming languages on Stack Overflow, calculated using equation 5, predicts the specialization of programming languages in Brazilian software data. The regression is run at the language-technical class level. Language-technical class pairs without Stack Overflow posts or registered software are assigned a value of zero in their respective measures. Both variables are normalized to have a mean of zero and standard deviation of one for ease of interpretation. Standard errors in parentheses. $^{*}p < 0.10,^{**} p < 0.05,^{***} p < 0.01$.

Figure A20: **Correlation Between Employment and the Instrument**



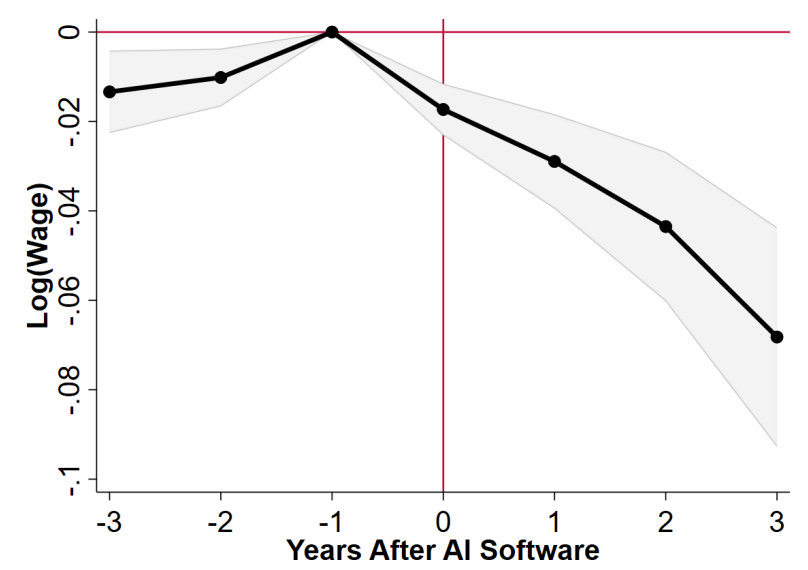
Notes: This figure plots the residual of log employment against the residual of the instrument. Observations are binned in 40 points.

Table A12: **AI Increases Hiring and Separation Rates**

	(1) *Hiring Rate*	(2) *Separation Rate*	(3) *Hiring + Separation Rates*
AI Exposure	0.0223**	0.0179*	0.0413*
	(0.0113)	(0.00997)	(0.0214)
Observations	43705	43435	43310

Notes: This table reports estimates of the effect of AI on hiring and separation, following the specification in equation (2). Column 1 is the hiring rate, defined as the number of hired workers divided by last period's employment; column 2 is the separation rate; and column 3 is the sum of the hiring and separation rates. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10,^{**}p < 0.05,^{***}p < 0.01$.

Figure A21: **Dynamic Effects of AI on Wages**



Notes: This figure reports estimates of the dynamic effect of AI on log wages, following the specification in Equation (3). All specifications use the AI ease-of-development instrument defined in Equation (6). Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors are clustered at the occupation level. The shaded area reports 95% confidence intervals.

Table A13: **Heterogeneous Effect of AI According to Intensity in Information Use**

	(1)	(2)	(3)	(4)	(5)	(6)
	log(Employment)	*log(Age)*	*log(Yrs. Education)*	*log(Tenure)*	*log(Wage)*	*log(Std. Wage)*
AI Exposure	−0.0928**	0.00332	0.00206	0.00939	0.00730	−0.0269
	(0.0366)	(0.00488)	(0.00420)	(0.0259)	(0.0130)	(0.0450)
AI Exposure×Info. Use	0.149***	−0.0110*	−0.0114**	−0.0388	−0.0401**	−0.0101
	(0.0465)	(0.00622)	(0.00547)	(0.0330)	(0.0165)	(0.0574)
N	45,336	45,336	45,332	45,336	45,336	45,208

Notes: This table reports 2SLS estimates of the effect of AI on labor market outcomes, allowing the parameter of interest to vary by interacting it with the relative information use intensity. Each column uses the AI ease-of-development instrument defined in Equation 6. In the first stage, the instrument is also interacted with the relative information use intensity. Column 1 reports the effect of AI on log employment; column 2 on the log of average age; column 3 on the log of average years of education; column 4 on the log of average tenure at the current firm (in years); column 5 on the log of average wage; and column 6 on the log of the standard deviation of wages. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to U.S. AI patents, and the lagged outcome variable. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10$,$^{**}p < 0.05$,$^{***}p < 0.01$.

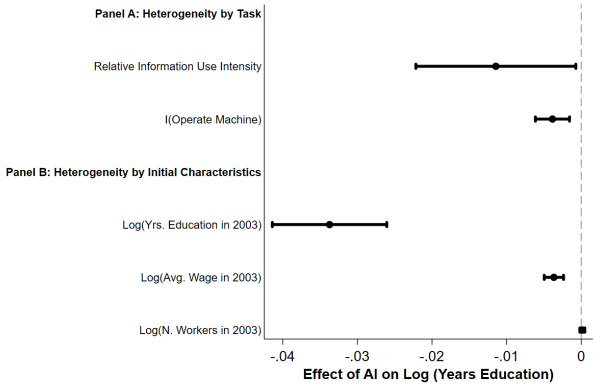
Table A14: **Effect of AI by Gender, Age, Education, and Experience**

	(1)	(2)	(3)
Group	*log(Employment)*	*log(Wage)*	*Ability*
By age			
Young *(less than 35)*	0.0212***	−0.0278***	−0.0872***
	(0.00801)	(0.00465)	(0.0156)
Old *(more than 35)*	0.0158**	−0.0298***	−0.0806***
	(0.00733)	(0.00461)	(0.0185)
By education			
High School or more	−0.0000127	−0.0266***	−0.0992***
	(0.00736)	(0.00418)	(0.0189)
High School incomplete	0.0260***	−0.0378***	−0.0767***
	(0.00901)	(0.00558)	(0.0181)
By experience			
$>$ *one year*	0.0175***	−0.0272***	−0.0874***
	(0.00757)	(0.00453)	(0.0230)
$\leq$ *one year*	0.0331***	−0.0408***	−0.0899***
	(0.0122)	(0.00632)	(0.0168)
By gender			
Male	0.0213***	−0.0236***	−0.0815***
	(0.00721)	(0.00436)	(0.0171)
Female	0.0240**	−0.0115**	−0.0442**
	(0.00957)	(0.00564)	(0.0196)

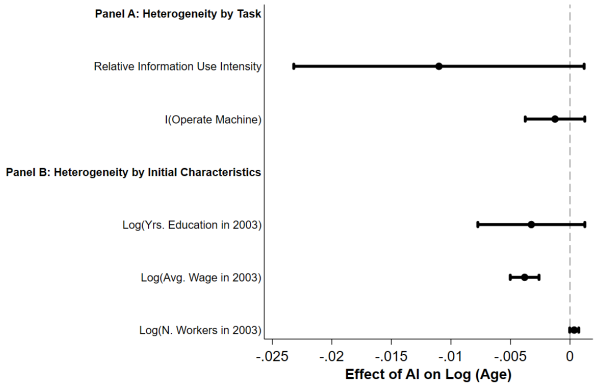
Notes: This table reports estimates of the effect of AI on employment and wages across demographic groups, following the specification in Equation (2). All specifications use the AI ease-of-development instrument defined in Equation 6. For each specification, the left-hand side variable is calculated among the sample described in *Group* column. Column 1 reports the effect of AI on log employment; column 2 on the log of average wages. Column 3 reports the effect of AI on ability. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to non-AI software, and exposure to U.S. AI patents. Standard errors, clustered at the occupation level, appear in parentheses. $^{*}p < 0.10$,$^{**}p < 0.05$,$^{***}p < 0.01$.

Figure A22: **Heterogeneous Effect of AI According to Tasks and Education**

(a) Heterogeneous Effect of AI on Years of Education



(b) Heterogeneous Effect of AI on Age



Notes: This figure reports heterogeneity in the effect of AI based on different characteristics of each occupation. To construct it, I augment the baseline model by including an interaction term: $y_{o,t} = \beta AI\ Exposure_{o,t} + \beta^{hetero} AI\ Exposure_{o,t} \times x_o + \mu_o + \mu_t + X'_{o,t}\theta + \epsilon_{o,t}$, where x_o denotes an occupation-level characteristic. Figure A22a plots the interaction coefficient β_x^{hetero} using log years of education as the outcome, while Figure A22b uses log age. Panel A uses task-based indicators constructed from occupational descriptions: *I(Operate Machine)* equals one if the occupation involves operating machinery in sectors like manufacturing, agriculture, or construction; *I(Planning or Execution)* equals one for roles involving process planning or execution, typical of high-level managers; *I(Management)* takes one for occupations managing people or information; *I(Interpersonal)* identifies occupations requiring interpersonal interactions, such as customer service or supervision; and *I(Creation and Innovation)* takes one for creative or innovation-related work. Panel B calculates heterogeneity using characteristics of the occupation measured at the beginning of the period. All specifications use the AI ease-of-development instrument defined in Equation 6. Controls include occupation fixed effects, one-digit-occupation-by-year fixed effects, exposure to non-AI software, and exposure to U.S. AI patents. Shaded areas represent 95% confidence intervals. Standard errors are clustered at the occupation level.